

Herramienta configurable para la evaluación y autoevaluación de la docencia virtual

Guillermo Martín Duque

Yuan Xu Jiang

Grado en ingeniería informática, Universidad Complutense de Madrid



Trabajo de fin de grado en informática

Madrid, 14 de junio de 2016

Director: Dº Antonio Sarasa Cabezuelo

Autorización

Guillermo Martín Duque y Yuan Xu Jiang, alumnos del Trabajo Fin de Grado (TFG) en el grado en Ingeniería Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TFG) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

TÍTULO del TFG: Herramienta configurable para la evaluación y autoevaluación de la docencia virtual

Curso académico: 2015/2016

Alumnos:

Fdo: Guillermo Martín Duque

Fdo: Yuan Xu Jiang

Fdo:Tutor del TFG: Antonio Sarasa Cabezuelo

Agradecimientos

En primer lugar queremos dar las gracias a nuestros familiares y amigos por su apoyo, ya que sin ellos no habríamos llegado hasta aquí.

Por otro lado dar las gracias a nuestro director de proyecto D^o Antonio Sarasa Cabezuelo, por su dedicación y colaboración como profesor y director en nuestro trabajo de fin de grado. Destacar que siempre ha estado a nuestra disposición cuando le hemos necesitado.

Y por último a la profesora D^a Ana María Fernández-Pampillón Cesteros, por ayudarnos a enfocar nuestra aplicación, proporcionándonos los requisitos que han sido necesarios. También por su colaboración junto a nuestro director en la elaboración de algoritmos, para así poder obtener algunas de las preguntas automáticas, por probar la aplicación, darnos los consejos a nivel de usuario y por toda su ayuda y dedicación.

Índice

Índice de figuras.....	3
Índice de abreviaturas.....	6
Resumen	7
Palabras clave.....	7
Abstract.....	8
Keywords	8
CAPÍTULO 1: Introducción	9
Introducción.....	9
Introduction	10
Objetivos del trabajo	11
CAPÍTULO 2: Especificación de la aplicación	12
2.1 Actores de la aplicación.....	12
2.2 Casos de usos	12
2.3 Descripción de la aplicación	16
CAPÍTULO 3: Tecnologías empleadas	44
3.1 Laravel: Framework PHP	44
3.2 XAMPP: Servidor web.....	44
3.3 MySQL: Base de datos	44
3.4 HTML5, CSS y Bootstrap	45
CAPÍTULO 4: Arquitectura de la aplicación.....	46
4.1 Aplicación Cliente - Servidor	46
4.2 Estructura Laravel	48
CAPÍTULO 5: Modelo de datos	50
CAPÍTULO 6: Diseño e implementación	65
6.1 Seguridad	65

6.2 Envío de correos electrónicos	68
6.3 Recuperación de la contraseña	68
6.4 Activación de una cuenta creada anteriormente	69
6.5 Descargar el resultado en PDF	69
6.6 Separar el código que puede variar con facilidad	70
6.7 Organización de las vistas	70
6.8 Estructuración mediante plantillas	71
6.9 Laravel Collective	73
CAPÍTULO 7: Procesamiento automático de la información.....	75
CAPÍTULO 8: Conclusiones y trabajo futuro	80
CONCLUSIONES	80
CONCLUSIONS	80
Trabajo futuro	81
Trabajo realizado por los miembros	81
Bibliografía	86
Anexo 1: Guía de instalación en el servidor	88
Anexo 2: Guía de instalación en Localhost	93
Anexo 3: Guía de uso de la aplicación	101
Glosario	124

Índice de figuras

Figura 1 Diagrama caso de uso – Profesor.....	13
Figura 2 Diagrama caso de uso – Evaluador.....	14
Figura 3 Diagrama Caso de Uso – Administrador.....	15
Figura 4 Página de inicio.....	16
Figura 5 Página de registro.....	17
Figura 6 Página de bienvenida.....	18
Figura 7 Gestión de perfil.....	19
Figura 8 Agregar profesor manualmente.....	24
Figura 9 Diagrama de araña.....	27
Figura 10 Pantalla de formas de ver el resultado.....	29
Figura 11 Visualización estática.....	29
Figura 12 Visualización dinámica	30
Figura 13 1ª y 2ª gráfica	31
Figura 14 3ª y 4ª gráfica	32
Figura 15 Visualización mediante consulta.....	34
Figura 16 Comparación del resultado de forma estática	35
Figura 17 Comparación del resultado de forma dinámica	36
Figura 18 Captura del resultado en PDF	37
Figura 19 Filtros en las tablas	38
Figura 20 Correo de confirmación de cuenta.....	39
Figura 21 Notificación por correo para restablecer la contraseña.....	39
Figura 22 Formulario para cambiar la contraseña	40
Figura 23 Notificación por correo para dar de alta una cuenta	41
Figura 24 Botón para dar de alta la cuenta.....	41
Figura 25 Resultado Académico	42
Figura 26 Gráficas del porcentaje de actividades y de participación de alumnos	43

Figura 27 Ejemplo de Cliente-Servidor	46
Figura 28 routers.php.....	47
Figura 29 Ejemplo Cliente-Servidor en nuestra aplicación	47
Figura 30 Modelo-Vista-Controlador	48
Figura 31 Users-Migration.....	50
Figura 32 Users-Migration.....	52
Figura 33 Password Resets_resets-Migration	53
Figura 34 Asignaturas-Migration	54
Figura 35 VersionesMoodle-Migration	55
Figura 36 Evaluaciones-Migration	56
Figura 37 Solicitudes-Migration.....	57
Figura 38 Preguntasautomaticas-Migration	58
Figura 39 Moodle-Migration	59
Figura 40 Formulario-Migration	60
Figura 41 Criterios-Migration	61
Figura 42 Boletin-Migration.....	62
Figura 43 Resumen de la base de datos.....	63
Figura 44 Model User	64
Figura 45 RegistroRequest.....	66
Figura 46 Administrador-Middleware	67
Figura 47 Código para enviar un correo	68
Figura 48 Código de nuestra aplicación	70
Figura 49 home.blade.php	72
Figura 50 template.blade.....	72
Figura 51 style.blade.....	73
Figura 52 Vista donde se sube el formulario.	89
Figura 53 Vista donde subir los criterios de evaluación.	90

Figura 54 Página donde se agregan las versiones Moodle.....	91
Figura 55 Instalar XAMPP.....	93
Figura 56 Instalación de XAMPP 2.....	94
Figura 57 Instalación XAMPP: indicar la ruta	94
Figura 58 Instalar COMPOSER	95
Figura 59 Instalar COMPOSER 2	96
Figura 60 Instalar COMPOSER 3.....	96
Figura 61 Archivo descomprimido en la carpeta tfg5	97
Figura 62 Consola con la ruta a la carpeta tfg5.....	98
Figura 63 Arrancar Apache y MySQL	99
Figura 64 Navegador con http://localhost/tfg5/public/login en la URL	100
Figura 65 Recuperar contraseña - ¿Olvido la contraseña?.....	102
Figura 66 Dar de alta una cuenta	103

Índice de abreviaturas

AJAX: Asynchronous JavaScript and Xml.

APP: Application Software.

BD: Base de datos.

CSS: Cascading Style Sheets.

HTML: Hypertext Markup Language.

HTTP: Hypertext Transfer Protocol.

IP: Internet Protocol.

JSON: JavaScript Object Notation.

MVC: Modelo Vista Controlador.

PDF: Portable Document Format.

PHP: Hypertext Preprocessor.

SQL: Structured Query Language.

URL: Uniform Resource Locator.

WAMPP: Windows, Apache, MySQL, PHP.

.

Resumen

En este proyecto se ha creado una aplicación web destinada a la valoración de la calidad de la docencia de asignaturas virtuales. Para ello se utiliza un modelo de evaluación que mide diferentes aspectos o factores que intervienen en la calidad. Este modelo se ha implementado en forma de un conjunto de cuestionarios que pueden ser utilizados tanto por profesores como por evaluadores. El profesor puede utilizarlo para obtener una autoevaluación de la calidad de la asignatura que ha impartido, y el evaluador lo usará en el contexto de comisiones de evaluación de la calidad de la docencia que se crean en las universidades y otras instituciones.

La evaluación se realiza de forma manual por el usuario, profesor o evaluador. Sin embargo existe un conjunto de factores de evaluación que son calculados de forma automática por el sistema tomando como entrada datos acerca de la asignatura virtual. El objetivo de esta aplicación no es sólo facilitar al usuario el formulario y obtener la puntuación, sino que también se obtenga de forma automática la puntuación de algunas preguntas y así poder completarlo de forma más cómoda y rápida.

Los resultados que muestra la aplicación son generados de forma dinámica, así el usuario podrá tener una mejor visión sobre la evaluación.

Palabras clave

Herramienta de evaluación de la calidad
Docencia online
Modelos de calidad e-learning
Aplicación web
Análisis de datos

Abstract

In this project it has developed a web application dedicated to assessing teacher's quality regarding virtual subjects. It has been used an evaluation model that assesses different aspects and factors that influence on how good is the teaching.

This model has been implemented through questionnaires that can be used by both professors and evaluators. Professors may use it to obtain a self-assessment of the quality of these subjects that has been taught and evaluators may make use of this model when it comes to evaluating the quality of teaching that are used in universities and similar institutions.

Evaluations are done without using technologies by users, teachers and evaluators. However there are a wide range evaluation factors that are calculated automatically by computer systems taking dates as virtual subjects. The aim of this application is not only make this type of questionnaires easy to use but also obtaining the score of some questions automatically and in this way it can be completed easier and faster.

In addition, the results are shown dynamically so that the users can have a better idea about the evaluation.

Keywords

Quality assessment tool
Online teaching
Quality e-learning models
Web Application
Data analysis

CAPÍTULO 1: Introducción

Introducción

En la actualidad todas las universidades realizan evaluaciones de calidad para conocer como fue el curso y así determinar en qué aspectos se necesita mejorar.

Existen diversas formas de realizar este tipo de encuesta, como puede ser mediante un papel o mediante alguna aplicación.

En la mayoría de las universidades se utilizan LMS (Learning Management Systems) que permiten disponer de una virtualización de las asignaturas presenciales y crear campus virtuales. Uno de los LMS más utilizados es [Moodle \[1\]](#). Que se caracteriza por su sencillez y facilidad de uso. Este tipo de herramientas guardan información de toda la actividad desarrollada en forma de [logs\[2\]](#) que pueden ser descargados para analizar sus datos.

La información que se puede obtener de los logs comprende datos acerca de quién ha visto un tema, quién ha consultado o escrito un comentario en el foro, que direcciones IP son las más habituales, quién ha realizado la encuesta docente,... etc.

Debido al gran número de información que se puede obtener y a la relevancia que tiene toda esta información, se pensó en explotar dicha información con el objetivo de evaluar la calidad de la docencia de las asignaturas virtuales.

Esta memoria se estructura de la siguiente manera. En primer lugar se plantean los objetivos del trabajo. Más tarde se describirá la especificación de la aplicación. A continuación se presentarán las tecnologías empleadas, la arquitectura de la aplicación y el modelo de datos. Posteriormente se describirá el diseño e implementación y el procesamiento automático de la información. Por último se establecerán las conclusiones y un conjunto de líneas de trabajo futuro.

Introduction

Nowadays, in all universities it is done evaluations of quality to know how the course was and determine what improvements have to be done.

There are several ways to carry out this type of survey, such as paper form, using some application ...etc.

In the majority of universities LMS (Learning Management Systems) are used and these systems allow users to have virtual subjects. There is a tool called [Moodle \[1\]](#) that it is used in almost all universities because it is easy to use and teachers and students can take advantage of this application. This tool saves all activities on a database that happen on [Logs \[2\]](#) and it can be downloaded in order to be analyzed.

Log's information is really valuable because with this kind of information it can be learnt things such as which person has watched a subject, who has seen and commented on a forum, which IP addresses are the most common, who has made the teaching survey, ... etc.

Due to the huge amount of information that can be obtained and the relevance of all this information it can be used with the aim of evaluating the teaching quality of these virtual subjects

This memory is organized in this way. Firstly the main objectives. Secondly the description of these objectives. After that the technology used and data models and how the application has been built.

Later application design will be described and implemented and automatic process of this information will be analyzed.

Finally conclusions will be reached and future work will be explained.

Objetivos del trabajo

En la actualidad la evaluación de la calidad de la formación ha tomado gran importancia, dado que la información que se puede extraer es clave para poder mejorar el proceso de aprendizaje. Este área tiene muchos aspectos que van desde la evaluación de los materiales utilizados, la metodología docente, la formación del docente, etc. Sin embargo hay un ámbito que es especialmente interesante controlar, que son las experiencias formativas online. Dada la naturaleza de las mismas, hay un mayor número de elementos que pueden ser medidos puesto que el docente se debe servir de diferentes herramientas para simular el proceso de aprendizaje que se realiza presencialmente. Así son típicos el uso de videos, transparencias, videoconferencias, espacios virtuales compartidos, etc. El tipo y el uso que hace el docente influyen en la experiencia y en los resultados académicos de los estudiantes. Este proceso puede ser medido, y es el objetivo de este trabajo fin de grado.

Partiendo de un modelo de evaluación de la calidad de la docencia virtual, en este trabajo se ha llevado a cabo la implementación de una aplicación web que automatiza su utilización por parte del profesor y del evaluador. Así, se plantearon los siguientes objetivos a alcanzar con este trabajo fin de grado:

1. Crear una aplicación web que implemente el modelo de calidad mencionado mediante formularios con preguntas destinadas a la valoración de la calidad de la docencia de una asignatura virtual de forma que pudiera ser utilizado por profesores o evaluadores.
2. Facilitar una visualización clara y dinámica de los resultados.
3. Evaluar de manera automática algunos aspectos de la calidad utilizando como datos de entrada los logs de Moodle

Para desarrollar el proyecto se realizó una planificación en 4 fases.

En la primera fase se llevó a cabo la especificación de la funcionalidad de la aplicación, y se analizaron las tecnologías que podrían utilizarse.

En la segunda fase se hizo el desarrollo del proyecto: diseño e implementación de la aplicación.

En la tercera fase se realizó una depuración de la aplicación, buscando posibles errores en la misma.

Por último, en la cuarta fase se ha llevado a cabo la redacción de la memoria.

CAPÍTULO 2: Especificación de la aplicación

2.1 Actores de la aplicación

En la aplicación se diferencian 3 actores:

1. **Profesor**
2. **Evaluador**
3. **Administrador**

1. El **profesor** tiene una lista de asignaturas que puede evaluar. Además de ser evaluadas por él. Una persona ajena, en este caso el evaluador, también podrá evaluarlas.
2. El **evaluador** se encargará de solicitar asignaturas de profesores para poder evaluarlas y que estos luego puedan verlas.
3. El **administrador** tendrá las siguientes funciones:
 - Se encargará de gestionar las solicitudes de cuentas de nuevos usuarios, tanto de profesores, evaluadores u otros administradores.
 - Confirmará o rechazará las solicitudes que planteen los evaluadores para poder evaluar la asignatura de un profesor.
 - Creará y borrará cuentas de los tres roles y creará y eliminará asignaturas.
 - Subirá el formulario de evaluación que desee, cambiará algunos criterios, como por ejemplo el rango de puntuaciones y añadirá nuevas versiones de Moodle.

2.2 Casos de usos

A continuación se van a mostrar los diferentes casos de uso definidos para el desarrollo de la aplicación:

a) Casos de uso referidos al profesor

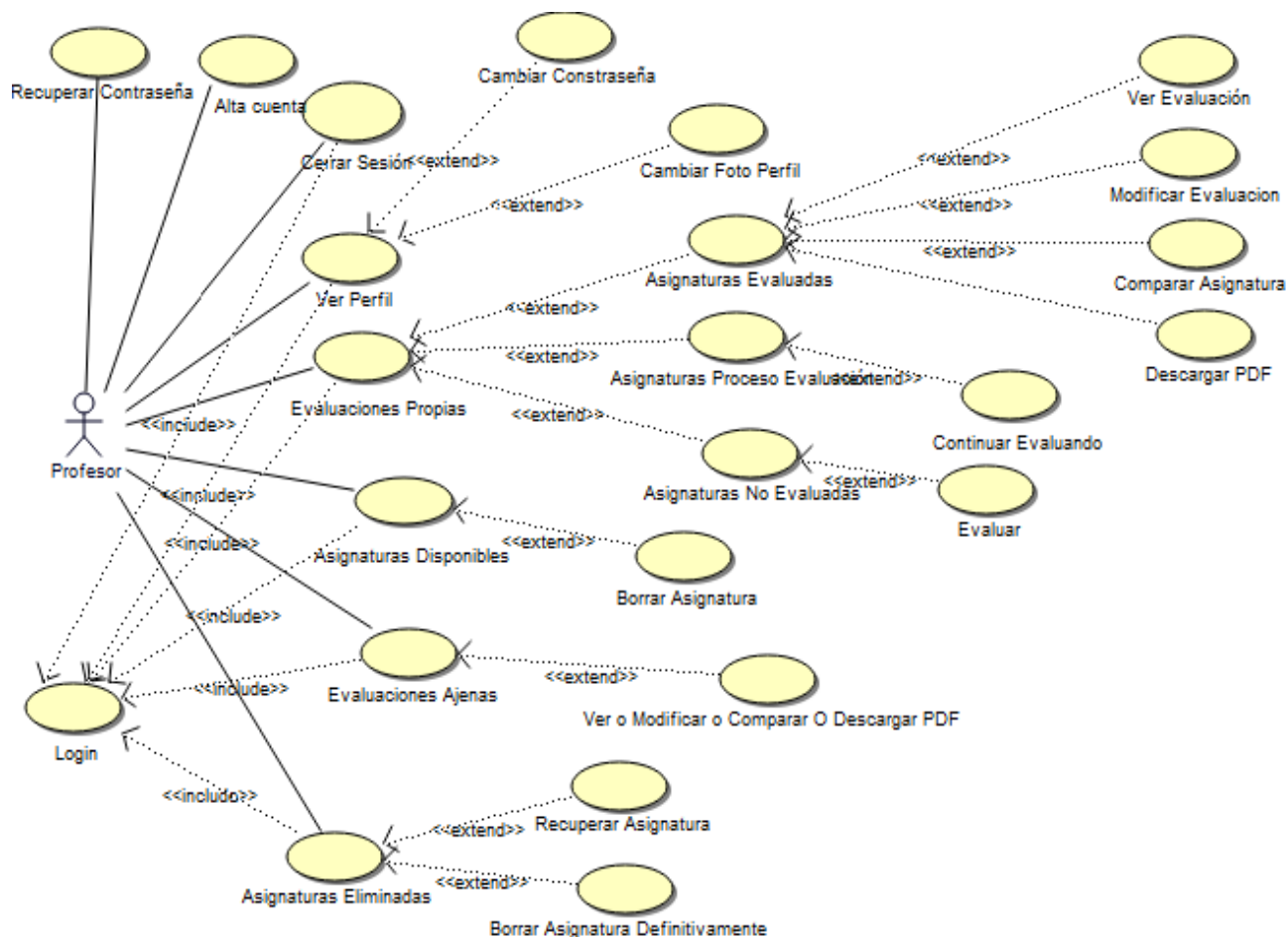


Figura 1 Diagrama caso de uso – Profesor

En la figura anterior se muestra el diagrama de caso de uso del Profesor.

En él se puede observar que para acceder a algunas partes, el usuario se tiene que identificar, por ejemplo para ver: perfil, asignaturas eliminadas, cerrar sesión, etc.

Por otro lado hay algunas acciones que no necesitan acreditación como: alta de cuenta o recuperar contraseña.

Además se muestra cómo son los menús en la aplicación.

Por último en este diagrama aparecen todas las acciones que el profesor puede realizar por ejemplo: evaluar una asignatura, eliminar una asignatura, etc.

b) Casos de uso referidos al evaluador

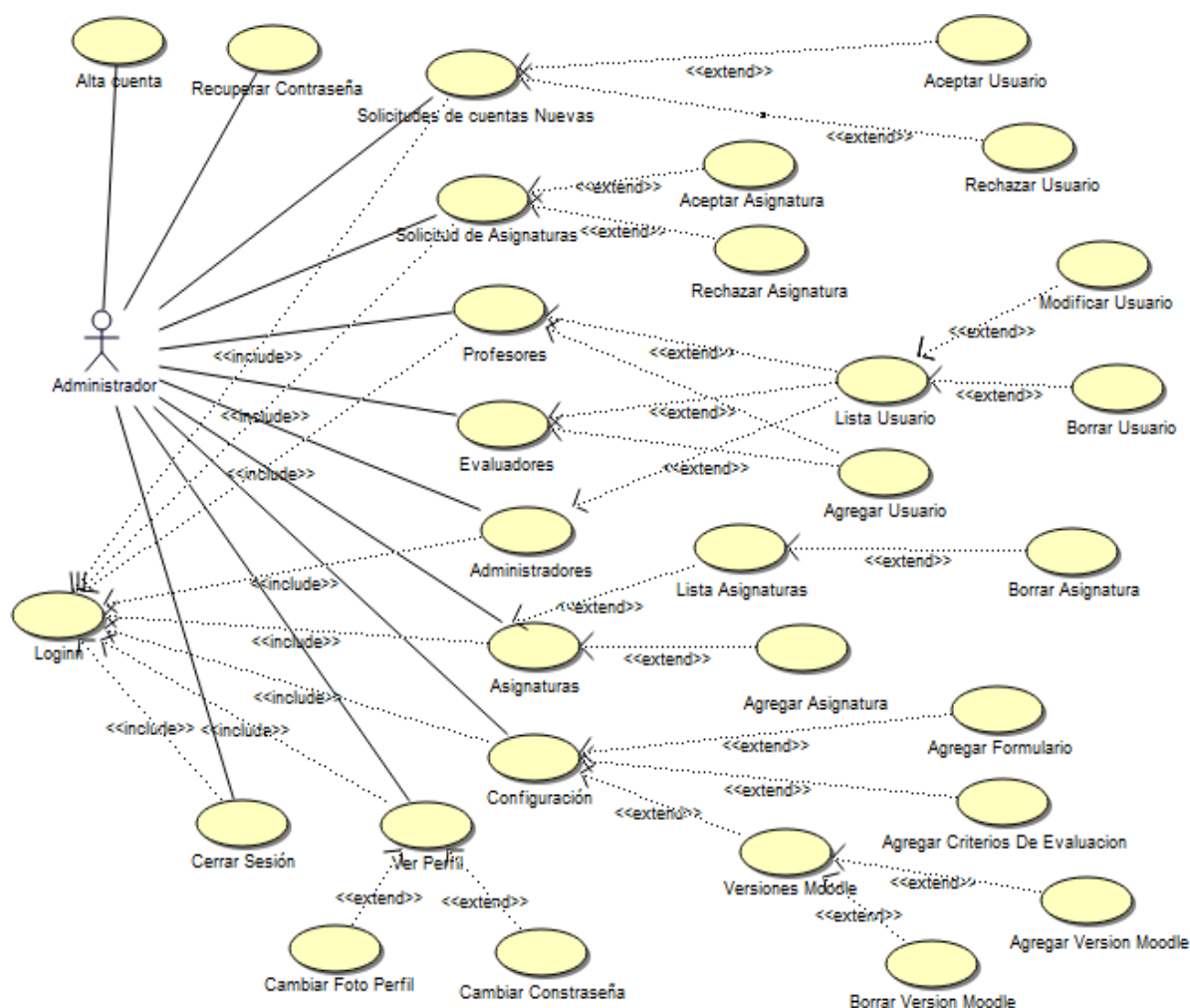


Figura 3 Diagrama Caso de Uso – Administrador

En la figura se muestra el diagrama de caso de uso del Administrador.

En este diagrama se observa que para ir algunas partes el usuario previamente se ha tenido que identificar, por ejemplo para: ver perfil, configuración, cerrar sesión, etc.

Por otro lado existen algunas acciones que no necesitan acreditación como: alta cuenta o recuperar contraseña.

Además se puede apreciar cómo son los menús en la aplicación.

Por último, destacar que en este diagrama aparecen todas las acciones que el administrador puede realizar como por ejemplo: agregar formularios, criterios de evaluación, etc.

2.3 Descripción de la aplicación

A continuación se describirán todas las partes que tiene esta aplicación explicando su funcionalidad.

2.3.1 Página de inicio

La primera vista que aparece de esta aplicación llamada [EVALUATE \[3\]](#) es la página de registro. Para poder acceder a la aplicación es necesario introducir el correo electrónico, la contraseña con la que se registro y el rol.

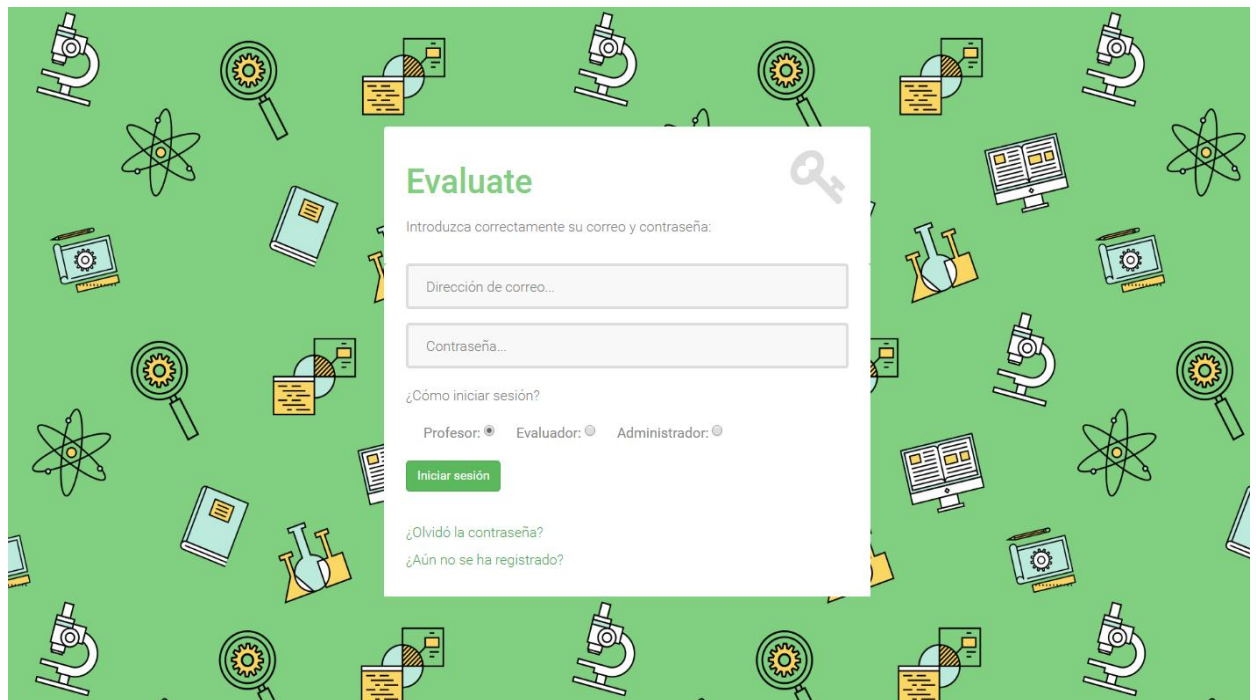


Figura 4 Página de inicio

En el caso de no estar registrado es posible hacerlo pinchando en:

- ¿Aún no se ha registrado?

Si se ha olvidado la contraseña es posible volver a recuperarla pinchando en:

- ¿Olvidó la contraseña?

2.3.2 Registro

En la página de registro se pide una serie de datos necesarios para la correcta identificación.

Regístrate

Introduzca los datos que aparecen a continuación para poder registrarse.

¿Con que rol quiere registrarse?

Profesor: ☒ Evaluador: ☐ Administrador: ☐

Nombre

Apellidos

DNI

Fecha de nacimiento

Figura 5 Página de registro

Es importante explicar algunos aspectos del funcionamiento del registro.

Un usuario con un mismo correo puede registrarse con los tres roles a la vez (profesor, evaluador y administrador). Ejemplo el usuario con el correo xuuyguille@gmail.com puede primero registrarse como administrador, luego más tarde como evaluador y por ultimo como profesor.

Las contraseñas con los tres roles bajo el mismo correo no tiene porque ser la misma.

Sin embargo un usuario con el mismo correo no puede registrarse dos veces bajo el mismo rol. Ejemplo: si el usuario con el correo xuuyguille@gmail.com se registra como profesor y más tarde vuelve a registrarse con el mismo correo y el mismo rol, se le notificara que su cuenta ya fue creada o solicitada anteriormente.

Por otro lado, es muy importante comentar que el administrador puede crear las cuentas de nuevos usuarios de cualquiera de los 3 roles. En este caso si el usuario intenta acceder con esa cuenta creada previamente por el administrador, la aplicación no le reconocerá la cuenta. En caso de que se registre, se le notificara que su cuenta con ese rol ya había sido creada anteriormente por el administrador y se le enviará un correo para poder activar la cuenta.

En caso de que el administrador no haya creado la cuenta previamente, el usuario una vez registrado, tendrá que esperar a que el administrador acepte su solicitud con esa cuenta para poder acceder. En el caso de que el administrador confirme la cuenta, se le enviara una notificación mediante correo de que su cuenta ha sido activada. Si la rechaza no tendrá notificación alguna por correo.

2.3.3 Menú principal

Lo primero que se muestra cuando se accede **EVALUATE** con cualquiera de los tres roles son tres regiones bien diferenciadas:

1. **Panel central**
2. **Panel superior**
3. **Menú lateral izquierdo**

1. En el **Panel central** aparece el contenido de la aplicación dependiendo de dónde se encuentre el usuario.
2. En **Panel superior** aparece la imagen del perfil y además contiene un menú en el cual se puede configurar el perfil y salir de la aplicación.
3. En el **Menú lateral izquierdo** hay varias opciones y estas varían en función del rol con el que se ha accedido.



Figura 6 Página de bienvenida

2.3.4 Configuración del perfil y cerrar sesión

Al pinchar sobre la imagen de perfil se desplegará un menú en el que si se pincha en cerrar sesión, se saldrá a la página de Login y se cerrará la sesión.

Al hacer click en configuración aparece una nueva vista en el panel central en la que se puede ver la información del usuario y además el usuario puede cambiar tanto la foto de perfil como la contraseña.

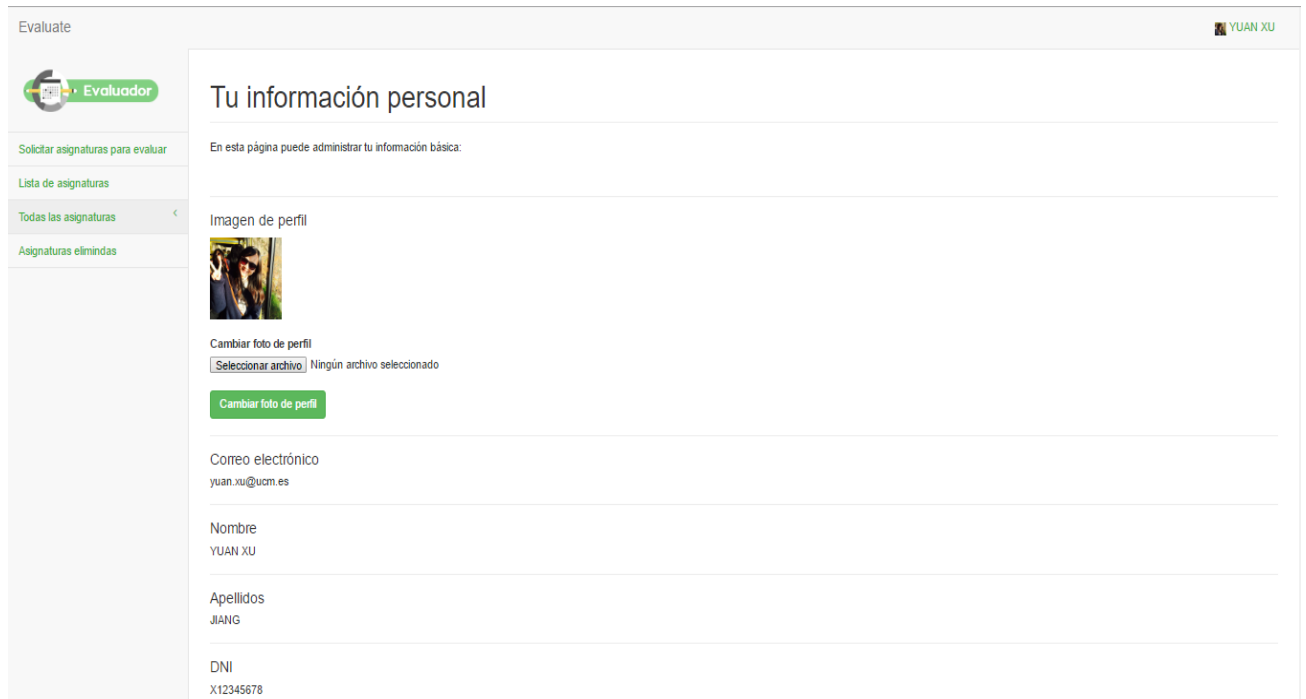


Figura 7 Gestión de perfil

2.3.5 Funcionalidad del Profesor

El profesor en el menú lateral izquierdo tienes varias opciones:

1. Asignaturas disponibles
2. Evaluaciones propias
3. Evaluaciones ajenas
4. Asignaturas eliminadas

1. **Asignaturas disponibles**, aparecerá una lista con todas las asignaturas que el profesor tiene pudiendo borrar las que no desee.

La funcionalidad de borrar que tiene el profesor sólo afecta a este y consiste en que una vez que elimine la asignatura no podrá visualizarla, es decir, posteriormente no podrá ni evaluarla, ni ver el resultado de su evaluación y solo le aparecerá en asignaturas eliminadas.

En ningún caso se eliminará de la base de datos por lo que el administrador y el evaluador podrán seguir trabajando con ella.

2. **Evaluaciones propias**, se desplegará un submenú con:

- **Asignaturas evaluadas**
- **Asignaturas en proceso de evaluación**
- **Asignaturas no evaluadas**

En **Asignaturas evaluadas** el profesor verá una lista con todas las asignaturas que él ha evaluado (finalizando definitivamente su evaluación), pudiendo ver, comparar con otra evaluación, descargar en PDF el resultado o modificar la evaluación.

En **Asignaturas en proceso de evaluación** el profesor verá una lista con las asignaturas que él ha empezado a evaluar, pudiendo continuar con su evaluación.

En **Asignaturas no evaluadas** se verán todas las asignaturas que el profesor aún no ha evaluado, e iniciar si lo considera oportuno dicho proceso.

3. **Evaluaciones ajenas**, se desplegará un submenú con las asignaturas que tiene el profesor, que han sido evaluadas o no por un evaluador:

- **Asignaturas evaluadas**
- **Asignaturas en proceso de evaluación**
- **Asignaturas no evaluadas**

En **Asignaturas evaluadas** el profesor verá una lista con todas las asignaturas evaluadas por un evaluador, pudiendo ver, comparar o descargar en PDF el resultado.

En **Asignaturas en proceso de evaluación** el profesor podrá ver que asignaturas ha empezado a evaluar un evaluador pero que aún no ha finalizado su evaluación.

Por último, en **Asignaturas no evaluadas** aparecerá una lista con las asignaturas que aún no han sido evaluadas por un evaluador.

4. **Asignaturas eliminadas**, el profesor podrá ver todas las asignaturas que haya borrado, pudiendo volver a recuperarlas o eliminarlas definitivamente.

El proceso de eliminar definitivamente sólo afecta al profesor y el resultado es que él nunca podrá ver esa asignatura.

En ningún caso se eliminará de la base de datos por lo que el administrador y el evaluador podrán seguir trabajando con ellas.

2.3.6 Funcionalidad del evaluador

El evaluador en el menú lateral izquierdo tienes varias opciones:

1. **Solicitar asignatura para evaluar**
2. **Lista de asignaturas**
3. **Todas las asignaturas**
4. **Asignaturas eliminadas**

1. **Solicitar asignatura para evaluar**, aparecerá un listado con todas las asignaturas que el evaluador puede solicitar. Esta petición será enviada al administrador, quien se encargará de aceptarla o rechazarla.

En el caso de que el administrador acepte la petición se le enviará un correo de notificación.

En caso contrario el evaluador podrá volver a solicitar la asignatura de nuevo.

2. **Lista de asignaturas**, se mostrará un listado con todas las asignaturas que tiene el evaluador, teniendo la opción de poder borrar las que considere.

El proceso de borrar consiste en que, al evaluador no le aparecerá la asignatura, por lo que no podrá ni evaluarla o ver su evaluación y sólo le aparecerá en asignaturas eliminadas.

Este proceso sólo afecta al evaluador, es decir, la asignatura no se borrará de la base de datos y tanto el administrador como el profesor podrán seguir trabajando con ella.

3. **Todas las asignaturas**, se desplegará un menú con los siguientes apartados:

- **Asignaturas evaluadas**

- **Asignaturas en proceso de evaluación**
- **Asignaturas no evaluadas**

En el apartado de las **Asignaturas evaluadas** se mostrará una lista con todas las asignaturas que ya han sido evaluadas, pudiendo ver, comparar, descargar en PDF su resultado o modificar la evaluación.

En el de **Asignaturas en proceso de evaluación**, el evaluador verá una lista con las asignaturas que él ha empezado a evaluar, pudiendo continuar con su evaluación.

Por último en el de **Asignaturas no evaluadas** se mostrará una lista con las asignaturas que no han sido evaluadas, teniendo la posibilidad de hacerlo.

En el caso de que el evaluador evaluara una asignatura, el resultado lo podrá visualizar inmediatamente el profesor de esa asignatura.

4. **Asignaturas eliminadas**, el evaluador podrá ver todas las asignaturas que haya borrado, pudiendo recuperar las asignaturas o eliminarlas definitivamente.

El proceso de eliminar definitivamente sólo afecta al evaluador y el resultado es que él nunca podrá ver esa asignatura.

En ningún caso se eliminará de la base de datos por lo que el administrador y el profesor podrán seguir trabajando con ella.

2.3.7 Funcionalidad del Administrador

El administrador en el menú lateral izquierdo tienes varias opciones:

1. **Solicitudes de cuentas nuevas**
2. **Solicitud de asignaturas**
3. **Evaluaciones**
4. **Profesores, Evaluadores o Administradores**
5. **Asignaturas**
6. **Configuración**
7. **Ranking**

1. **Solicitudes de cuentas nuevas**, el administrador podrá aceptar o rechazar las solicitudes de cuentas nuevas que tenga tanto de profesores, evaluadores y administradores. En caso de aceptarla se enviará un correo automáticamente de confirmación.

2. **Solicitud de asignaturas**, aparecerá una lista con todas las solicitudes de asignaturas que hayan hecho los evaluadores. El administrador podrá aceptarlas o rechazarlas. En caso de aceptarlas se enviará un correo automáticamente de confirmación.
3. **Evaluaciones**, el administrador podrá ver las evaluaciones de los profesores y evaluadores y descargar su resultado en PDF.

4. **Profesores, Evaluadores o Administradores**, en este apartado se desplegará un submenú con dos apartados:

- **Lista**
- **Agregar**

En **Lista** el administrador podrá ver todos los profesores, evaluadores o administradores permitiéndole poder modificar y borrar al que desee.

El proceso de borrar consiste en eliminar de la base de datos a usuarios por lo que si un usuario es borrado, no podrá acceder a la aplicación.

Si el usuario borrado es un profesor, además de borrar a ese usuario, se eliminarán también todas sus asignaturas asociadas. En el caso de que alguna de estas asignaturas estuviese siendo evaluada por un evaluador, este no podrá visualizarla tampoco.

Si el evaluador es borrado y había evaluado alguna asignatura, el profesor no podrá ver esa evaluación.

En **Agregar** el administrador podrá crear cuentas de profesores, evaluadores y otros administradores.

Existen dos formas de crear un nuevo usuario (administrador, profesor o evaluador):

- **Mediante la subida de un fichero**
- **Mediante un formulario.**

En el primer caso el administrador tendrá que subir un archivo Excel con formato CSV (delimitado por comas), según se explica.

En el segundo caso tendrá que rellenar todos los datos según se piden en el formulario.

Agregar profesor manualmente

Para agregar un profesor debe rellenar todos los campos que aparecen a continuación.

Fecha de nacimiento

dd/mm/aaaa

Nombre...

Apellidos...

DNI...

Dirección de correo...

Crear profesor

Figura 8 Agregar profesor manualmente

- 5. Asignaturas**, el administrador podrá crear y borrar asignaturas. Además puede asignar directamente un evaluador a una asignatura.

Cuando se borra una asignatura lo que se hace es eliminarla de la base de datos, es decir, el profesor que tenía asociado esta asignatura o el posible evaluador, no podrán trabajar con ella ya que les desaparecerá para siempre.

Existen dos formas de crear una nueva asignatura:

- **Mediante la subida de un fichero**
- **Mediante un formulario**

Con la primera forma el administrador tendrá que subir un fichero Excel con formato CSV (delimitado por comas) según se indica.

Con la segunda forma el administrador tendrá que rellenar un formulario donde se pide el nombre de la asignatura, el grupo, el curso y el correo del profesor que la impartirá.

En el caso de que el profesor no haya sido registrado en la base de datos, se creará uno nuevo asignándole esa dirección de correo electrónico.

6. Configuración, este apartado sirve para obtener una aplicación flexible, que permita poder adaptarse en futuro a posibles cambios.

Estos cambios podrían ser:

Primero que el formulario de evaluación puede cambiar con el tiempo o que puede ser diferente dependiendo de la universidad.

Segundo, algunos datos o valores, como puede ser el rango de puntuación, pueden variar fácilmente.

Tercero, la versión de Moodle puede cambiar y con este cambio el formato de los Logs de Moodle.

Cuarto, el código que obtiene las preguntas automáticas puede variar al cambiar el formulario.

Por eso es necesario que el formulario se pueda cambiar fácilmente y que exista en la base de datos una tabla capaz de guardar ciertos valores que puedan cambiar con facilidad.

Es necesario también que en la base de datos haya una tabla, que en el caso de que el formulario contenga preguntas automáticas, sepa cuál es el script capaz de obtener esas preguntas.

Por último, que exista una tabla en la base de datos que tenga la ruta al script capaz de leer los Logs de Moodle de esa versión.

Teniendo en cuenta todo esto, se desplegará un submenú con tres apartados:

- **Formulario**
- **Criterios de evaluación**
- **Versiones de Moodle**

En **Formulario**, el administrador podrá subir un fichero txt con el formulario de evaluación que quiere que vean tanto los profesores como los evaluadores al evaluar una asignatura.

Además si ese formulario contiene alguna pregunta que se puede evaluar automáticamente, hay que indicar la ruta al script que contiene el código para obtener la puntuación automática.

Conviene destacar dos casos que se pueden producir:

Si el administrador no ha subido ningún formulario, tanto el profesor como el evaluador no podrán realizar la evaluación y la propia aplicación les avisará con un mensaje de que no pueden evaluar.

Si el administrador ha subido un formulario y un profesor o un evaluador han hecho una evaluación de una asignatura con ese formulario, si posteriormente el administrador cambia el formulario y sube uno nuevo, entonces cuando el profesor y el evaluador van a realizar una evaluación de otra asignatura, el cuestionario que les aparecerá será el último formulario que ha subido el administrador.

¿Qué sucede si el profesor y el evaluador quieren ver el resultado de la evaluación de la asignatura que contiene el primer formulario que se subió? En ese caso les aparecerá el resultado de la evaluación con el primer formulario.

Si el usuario ha hecho una evaluación de una asignatura con un formulario que no es el actual y luego hace otra evaluación de otra asignatura con el formulario actual, si posteriormente quiere comparar el resultado de ambas evaluaciones dinámicamente, no podrá y la propia aplicación le avisará con un mensaje.

Si se compara el resultado de ambas evaluaciones estáticamente, no habrá ningún problema y se podrá realizar.

En **Criterios de evaluación**, el administrador puede subir un archivo txt en el que se especifican algunos criterios para la evaluación, como cuál es el rango de puntuación de las preguntas. Automáticamente la aplicación se adaptará a los rangos establecidos, teniendo en cuenta las cosas que pueden variar, como puede ser calcular el resultado o mostrarlo.

En **Versiones Moodle** se desplegará un submenú con dos opciones:

- **Lista de versiones**
- **Agregar versiones**

En **lista de versiones** aparecen todas las versiones de Moodle que existen en la aplicación pudiendo eliminarlas.

En **agregar versiones** aparecerá un formulario para añadir una nueva versión.

Para añadir una nueva versión hay que indicar el nombre de la versión y la ruta al script que contiene el algoritmo para poder leer los logs de Moodle correspondientes y calcular los resultados automáticos de esa versión.

7. **Ranking**, en este apartado el administrador podrá ver una gráfica que recoge todas las evaluaciones tanto de profesores como de evaluadores haciendo una media teniendo en cuenta los apartados y los ordena.

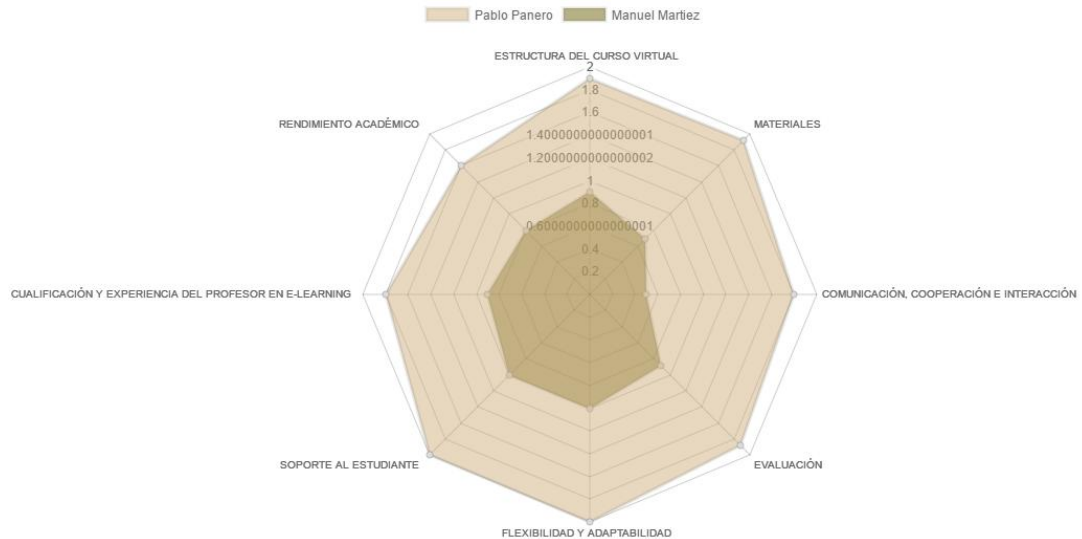


Figura 9 Diagrama de araña

2.3.8 Formas de evaluar

Tanto los profesores como los evaluadores tienen dos formas posibles de evaluar una asignatura:

1. **Manualmente**
2. **Automática**

1. **Manualmente** implica que ninguna pregunta se valorará automáticamente, es decir, el usuario tendrá que introducir la puntuación de forma manual.
2. **Automática** significa que algunas preguntas del formulario se puntuarán de forma automática sin que el usuario tenga que hacer nada. Aunque sino está de acuerdo se le permitirá modificarlo.

Al seleccionar esta última forma de evaluar automáticamente, el usuario obligatoriamente tendrá que subir previamente dos archivos, uno que contenga el log del Moodle y otro con el boletín de notas. El formato de estos dos archivos pueden ser o bien un archivo Excel CSV (delimitado por comas) o preferiblemente un archivo txt delimitado por “;”.

El archivo del boletín de notas debe contener los siguientes datos:

- **Nombre**
- **Apellido**

- ID
- ID_USUARIO
- DNI/NIF
- Dirección de correo
- Curso procedencia
- ¿Superó la evaluación continua?
- Tareas examen final
- Total del curso

Todos estos campos, menos ¿Superó la evaluación continua?, aparecen cuando se descarga el boletín de notas del Moodle.

El campo, ¿Superó la evaluación continua?, indica si el alumno supero la asignatura mediante la evaluación continua. Y puede tener dos valores: sí y no. “Si” en el caso de que el alumno haya superado la evaluación continua y “No” en el caso contrario.

Por último, el usuario debe indicar la versión de Moodle que está utilizando.

Una vez cumplido estos requisitos y tras hacer click en el botón Evaluar automáticamente, aparecerá el formulario con la puntuación en cada pregunta automática. Esta puntuación podrá ser modificada por el usuario si no se está de acuerdo.

Es importante destacar que en los dos métodos (automático y manual), en el caso de que se finalice una evaluación pero no se ponga la puntuación en alguna pregunta, estas serán calificadas como NV (no verificable) y también en el caso de introducir una puntuación incorrecta, fuera del rango de puntuaciones, será calificada como NV.

2.3.9 Formas de visualizar el resultado

Tanto los profesores como los evaluadores van a tener tres formas para poder ver el resultado:

- 1. Estática**
- 2. Dinámica**
- 3. Mediante consulta**



Figura 10 Pantalla de formas de ver el resultado

1. En la **visualización estática** se verá el resultado en el propio cuestionario.

Resultado de la evaluación

A continuación se mostrará el resultado de la evaluación:

Deportes	
Pregunta	Puntuación (0-1-2-3-NA-NV)
1. El curso virtual se estructura en m?dulos autocontenidos.	1
2. Los modulos deben tener una coherencia interna.	2
3. El mundo es azul.	3
4. Los estudios	0
PUNTUACION TOTAL Deportes :	6 / 12 ----> 50%

Técnicas	
Pregunta	Puntuación (0-1-2-3-NA-NV)
1. El curso virtual se estructura en m?dulos autocontenidos.	3
PUNTUACION TOTAL Técnicas :	3 / 3 ----> 100%

Puntuación final: 9 / 15 ----> 60%

Figura 11 Visualización estática

2. En la **visualización dinámica** se verá el resultado en forma de gráficas.

En esta pantalla existe la posibilidad de dos opciones:

- **Ver el resultado global de la evaluación**
- **Ver el resultado de un apartado concreto que seleccionemos**



Figura 12 Visualización dinámica

En las dos opciones aparecen cuatro graficas:

1. La primera gráfica, **Resultado general en %**, muestra la puntuación total que se ha obtenido en todos los apartados y la puntuación que falta para completar el 100% de la nota, en el caso de que no se haya completado.
2. La segunda gráfica, **Resultado en cada apartado en % (el máximo que se puede obtener en cada apartado es 100%)**, muestra la puntuación en % que se ha obtenido en cada apartado. La máxima puntuación que se puede obtener en cada apartado es el 100%.

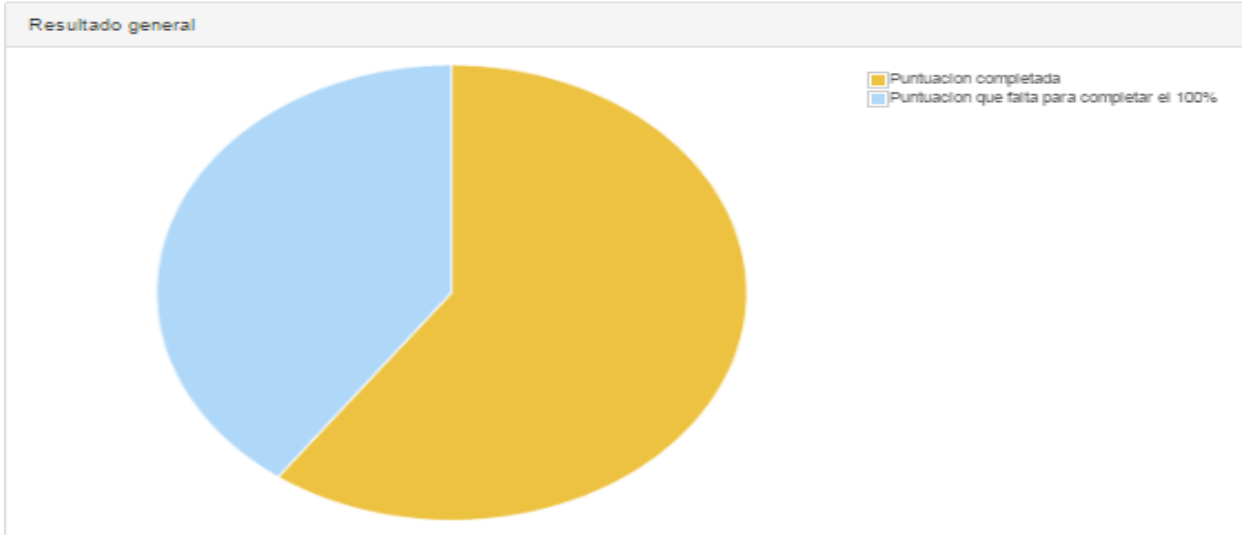
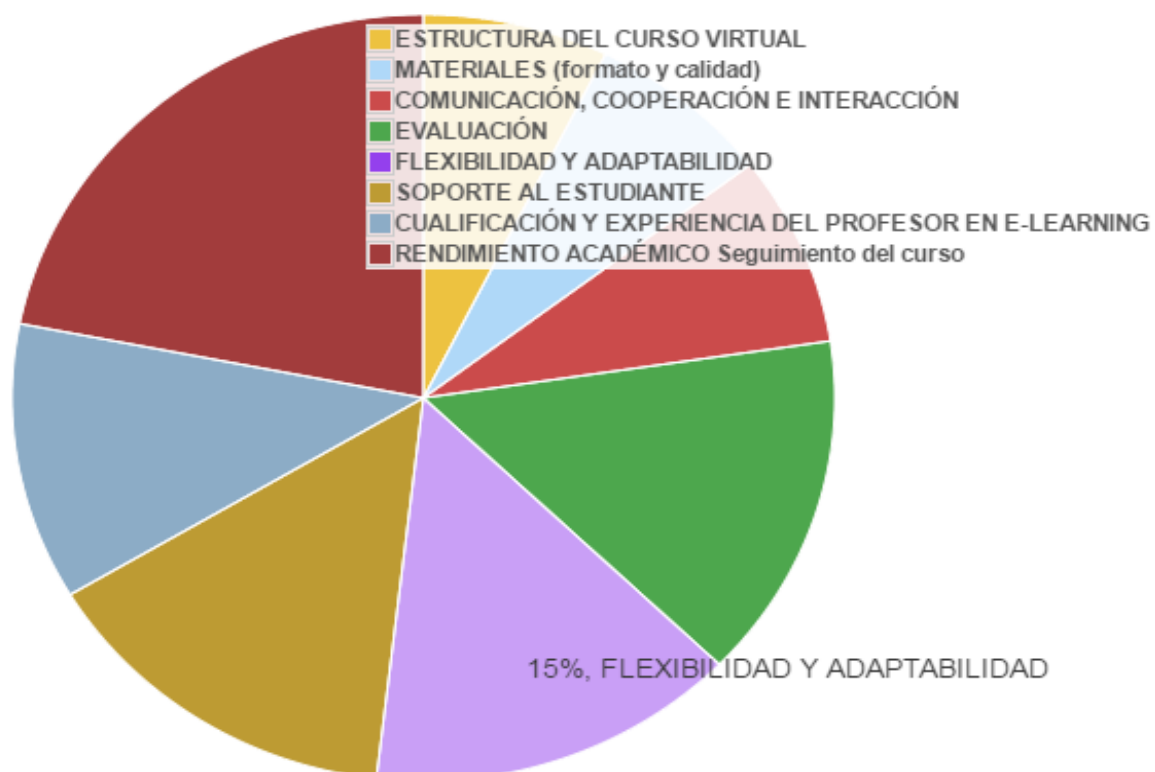


Figura 13 1ª y 2ª gráfica

- La tercera gráfica, **Resultado en cada apartado en % con gráfica de quesito**, representa lo mismo que la gráfica **Resultado en cada apartado (el máximo que se puede obtener en cada apartado es 100)** pero con forma de queso.

Resultado en cada apartado con gráfica de quesito



Puntos obtenidos en cada apartado

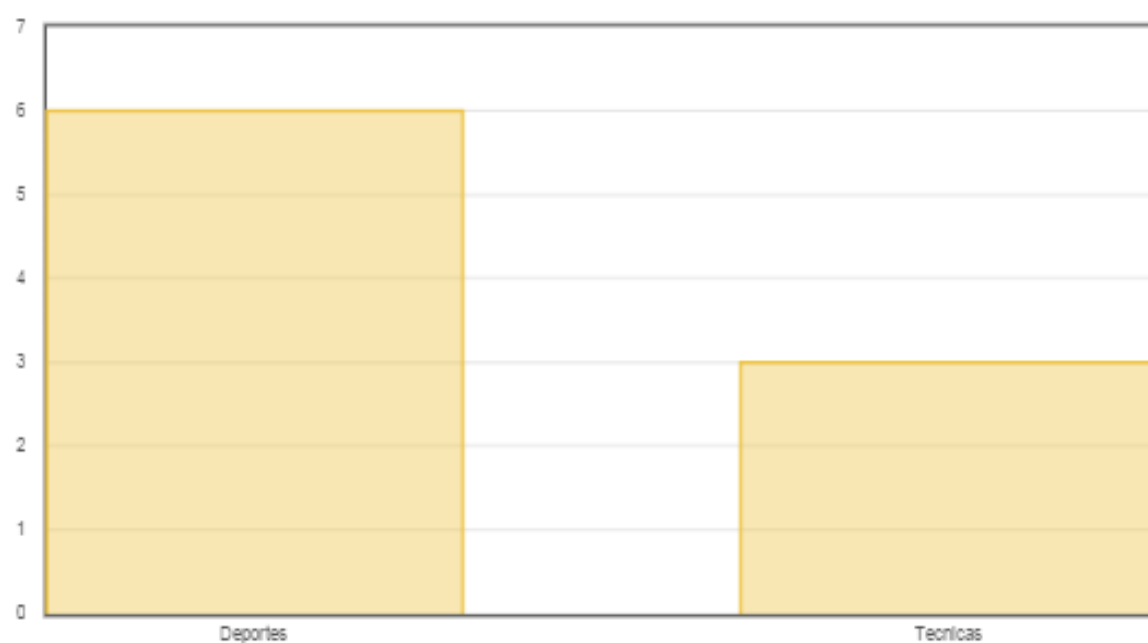


Figura 14 3ª y 4ª gráfica

4. La cuarta gráfica, **Puntos obtenidos en cada apartado**, muestra la puntuación total que se ha obtenido en cada apartado.

A continuación se va a mostrar un ejemplo para poder entender mejor el funcionamiento de las gráficas.

Se va a suponer que este ha sido el resultado de la evaluación de una asignatura:

Estructura del curso virtual

Pregunta 1 -> puntuación 3

Pregunta 2 -> puntuación 2

Pregunta 3 -> puntuación NV

Materiales

Pregunta 1-> puntuación 2

Pregunta 2-> puntuación 0

Y el rango máximo de cada pregunta es 3.

En el caso de que exista una puntuación NV (no evaluable), esa puntuación no se sumaría.

Así en el apartado Estructura de curso virtual la puntuación máxima de este ejemplo sería por las preguntas 1 y 2 en total 6 y la pregunta 3 no se tiene en cuenta porque es NV.

La gráfica **Resultado en cada apartado en % (el máximo que se puede obtener en cada apartado es 100%)**, lo que hace esta gráfica en este ejemplo es:

Obtener la media en cada apartado, es decir, sumar la puntuación de todo el apartado y se divide entre la puntuación máxima de ese apartado.

Estructura del curso virtual: $(3+2)/6 = 5/6 = 83,33\%$

Materiales $(2+0)/6 = 2/6 = 33,33\%$

Por lo que la gráfica mostraría dos barras: estructura del curso virtual con un valor de 83,33% y la de materiales con un valor de 33,33%.

En la gráfica **Resultado en cada apartado en % con gráfica de quesito**, lo que se hace es introducir los valores anteriores (83,33%, 33,33%).

La gráfica **Resultado general en %**, lo que haría en este ejemplo, es sumar todos los apartados y luego dividirlo por la puntuación máxima de todos los apartados.

En este ejemplo el algoritmo funcionaría de la siguiente manera:

$$5/6 + 2/6 = 7/12 = 58\%$$

Y lo que muestra es:

Puntuación completada: 58%

Puntuación para completar el 100% (100-58) = 42%

Por último la gráfica **Puntos obtenidos en cada apartado**, mostraría:

Una gráfica con dos campos:

- Estructura del curso virtual
- Materiales

El campo de Estructura del curso virtual tendrá un valor de 5, (3+2) y en el campo de Materiales un valor de 2, (2+0).

3. En la **visualización mediante consulta** aparecerá una lista para poder seleccionar el apartado del formulario que se desea consultar y posteriormente aparecerán las preguntas de ese apartado pudiendo seleccionar aquellas que se quiera conocer el resultado.



Pregunta	Puntuación
1. El material está bien seleccionado	3
2. El material se presenta con apoyos gráficos y/o sonoros cuando sea posible y para favorecer la comprensión y acceso al contenido.	3

Figura 15 Visualización mediante consulta

Se pueden consultar varias preguntas de un mismo apartado y en caso de que se seleccione una pregunta que no exista la aplicación informará de que no existe.

2.3.10 Comparar el resultado

Al usuario le puede resultar muy útil comparar el resultado de la evaluación de una asignatura con otra.

La manera de visualizarlo es la misma que si el usuario ve una evaluación por separado, lo único que la pantalla se divide en dos partes:

- **A la izquierda la evaluación de una asignatura**
- **A la derecha la evaluación de la otra asignatura**

De esta forma el usuario puede comparar el resultado de ambas asignaturas de forma más cómoda.

Evaluate

Manuelito

Profesor

Asignaturas disponibles

Evaluaciones propias <

Evaluaciones ajenas <

Asignaturas eliminadas

Nombre: EDA

Grupo: 1ºB

Curso: 2014-2015

Evaluada por: guimarti@ucm.es

A continuación se mostrara el resultado de la evaluación:

ESTRUCTURA DEL CURSO VIRTUAL	
Pregunta	Puntuación (0-1-2-NA-NV)
1. El curso virtual se estructura en m	3
2. Los modulos deben tener una coherencia interna.	2
3. El kskfkfkfkfk...	1
PUNTUACION TOTAL ESTRUCTURA DEL CURSO VIRTUAL :	6 / 9 ----> 66.666666666667%

Nombre: MMI

Grupo: 1ºB

Curso: 2012-2015

Evaluada por: guimarti@ucm.es

A continuación se mostrara el resultado de la evaluación:

ESTRUCTURA DEL CURSO VIRTUAL	
Pregunta	Puntuación (0-1-2-NA-NV)
1. El curso virtual se estructura en m	1
2. Los modulos deben tener una coherencia interna.	3
3. El kskfkfkfkfk...	1
PUNTUACION TOTAL ESTRUCTURA DEL CURSO VIRTUAL :	5 / 9 ----> 55.555555555556%

Figura 16 Comparación del resultado de forma estática

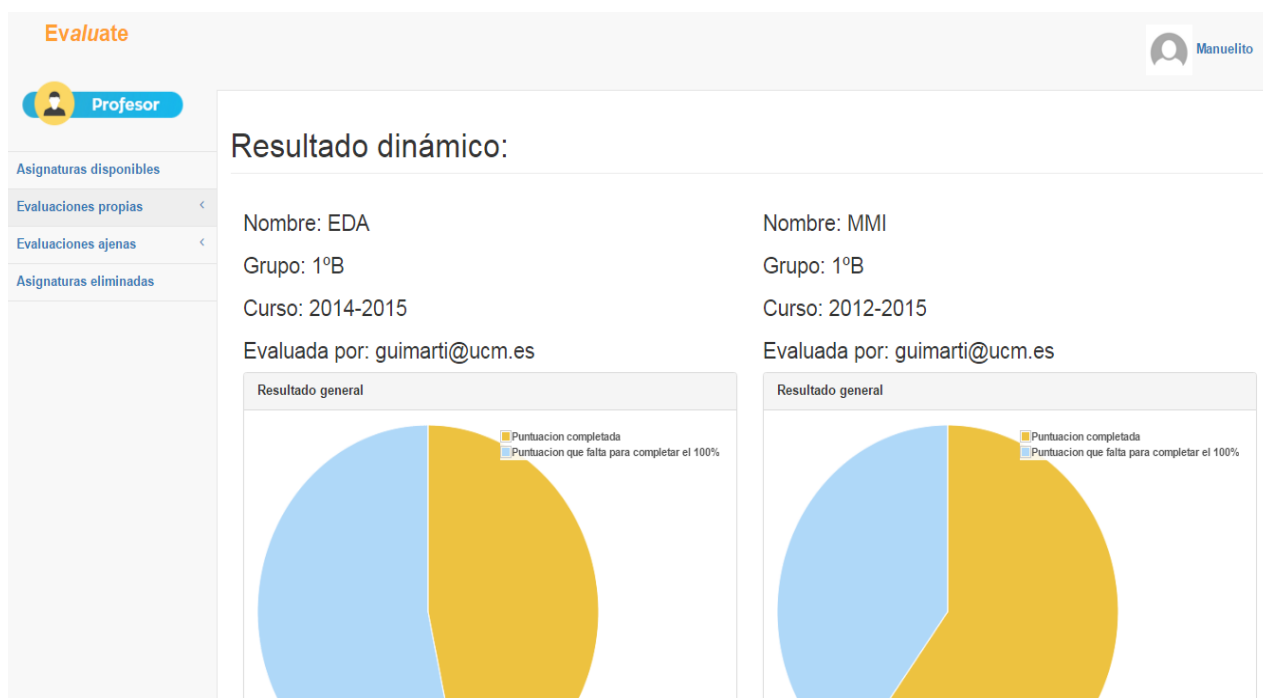


Figura 17 Comparación del resultado de forma dinámica

2.3.11 Descargar el resultado en PDF

D^a Ana María Fernández-Pampillón Cesteros comentó que sería muy útil para los profesores, poder descargar el resultado en PDF.

Por tanto una de las funcionalidades que tiene esta aplicación es que se puede descargar el resultado en PDF.

ESTRUCTURA DEL CURSO VIRTUAL

1. El curso virtual se estructura en módulos autocontenidos (organizados por semanas o temas). **Puntuación: 1**
 2. Los módulos deben tener una coherencia interna **Puntuación: 1**
 3. Los módulos deben estar bien ordenados de forma que se preserve la coherencia del conjunto y formen uno o varios "itinerarios de aprendizaje" **Puntuación: 1**
 4. Existe coherencia entre la estructura del curso virtual (el diseño de aprendizaje) y la metodología didáctica que utiliza el profesor. **Puntuación: 1**
 5. El curso virtual y cada uno de sus módulos dispone de una ficha donde se especifican: los objetivos de aprendizaje, tiempo de trabajo requerido (horas), guía para el aprendizaje (guía de estudio), materiales didácticos básicos, materiales didácticos co **Puntuación: 1**
 6. La información se encuentran fácilmente en el espacio virtual **Puntuación: 1**
 7. La forma de comunicación entre alumnos y alumnos y profesor está organizada y es fácil **Puntuación: 1**
 8. El curso virtuales "amigable" y fácil de usar **Puntuación: 1**
 9. El curso virtual promueve la creatividad, el pensamiento crítico y el conocimiento profundo de la materia **Puntuación: 1**
 10. El profesor revisa y mejora el espacio virtual. Si existen, utiliza los resultados de la evaluación de los estudiantes sobre su experiencia de aprendizaje en la plataforma para mejorar el espacio virtual. **Puntuación: 1**
- Puntuación: final ESTRUCTURA DEL CURSO VIRTUAL : 10 / 30 -----> 33.33%**

Figura 18 Captura del resultado en PDF

2.3.12 Modificar la evaluación

La aplicación tiene la funcionalidad de poder modificar la evaluación ya que un profesor o un evaluador tras evaluar una asignatura necesitan o desean modificarla.

A la hora de volver a evaluar no se podrá utilizar el método de evaluación automática, tendrá que hacerlo de forma manual.

Al usuario le aparecerá el resultado de la evaluación pudiendo modificar la puntuación de forma manual.

2.3.13 Filtros en las tablas

Se pueden aplicar ciertos filtros en las tablas, gracias a la utilización de [DataTables \[4\]](#) que utiliza para su funcionamiento JavaScript y AJAX.

Show entries Search:

Nombre de la asignatura	Grupo	Curso	Acción a realizar
EDA	1ºB	2012	<button>Eliminar asignatura</button>
MMI	1ºB	2012	<button>Eliminar asignatura</button>
TP	1ºB	2012	<button>Eliminar asignatura</button>

Showing 1 to 3 of 3 entries

Previous 1 Next

Figura 19 Filtros en las tablas

Estos filtros permiten poder buscar por:

- **Nombre.**
- **Ordenar el resultado por orden alfabético según el campo de la tabla que se seleccione.**
- **Seleccionar el número de entradas que haya como máximo en la tabla.**

Todos estos filtros ayudan al usuario a poder encontrar y mostrar la asignatura de una manera mucho más rápida y cómoda.

2.3.14 Notificación mediante correos

La aplicación tiene la funcionalidad de poder enviar notificaciones mediante correos electrónicos.

Las notificaciones por correo se utilizan cuando el administrador:

- **Acepta una solicitud de una asignatura o de una nueva cuenta.**
- **Cuando un usuario quiere recuperar su contraseña.**
- **Para activar una cuenta creada previamente por el administrador.**

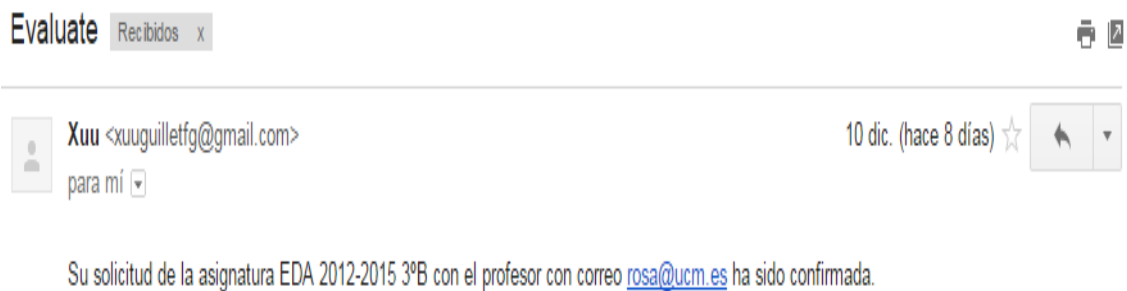


Figura 20 Correo de confirmación de cuenta

2.3.15 Recuperación de la contraseña

Los usuarios de cualquier rol ya sea profesor, evaluador o administrador pueden recuperar la contraseña en caso de que la hayan perdido.

Primero hay que ir a la página de inicio y pinchar en:

- ¿Olvido la contraseña?

Una vez realizado este paso aparecerá un formulario en donde hay que escribir la dirección de correo electrónico del usuario.

Accediendo a esta dirección de correo se podrá comprobar que la aplicación ha enviado un mensaje.

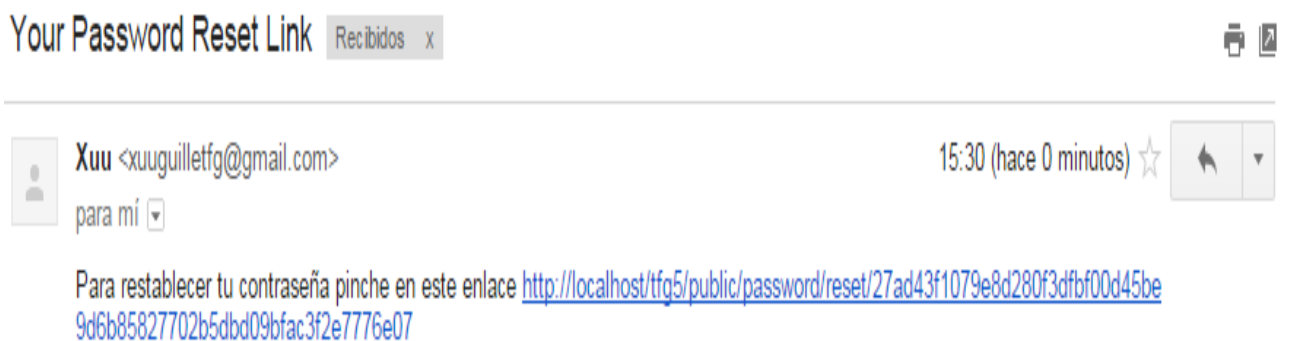


Figura 21 Notificación por correo para restablecer la contraseña

Para seguir con la recuperación de la contraseña hay que pinchar en el enlace que hay en el correo.

Después de pinchar en él aparecerá un formulario en el que se debe indicar cuál es el correo, rol y la nueva contraseña que se quiere establecer.

El formulario se presenta en un recuadro blanco con una sombra sobre un fondo verde con patrones de laboratorio. El título es '¿Olvidaste la contraseña?'. El texto de instrucción dice: 'Introduce la información que se solicita a continuación para poder recuperar tu contraseña.' A continuación, se pregunta '¿Con que rol quiere recuperar la contraseña?' con tres opciones de radio: 'Profesor' (seleccionada), 'Evaluador' y 'Administrador'. Hay tres campos de texto: 'Dirección de correo...', 'Contraseña...' y 'Repita la contraseña...'. Al final, hay un botón verde que dice 'Restablecer la contraseña'.

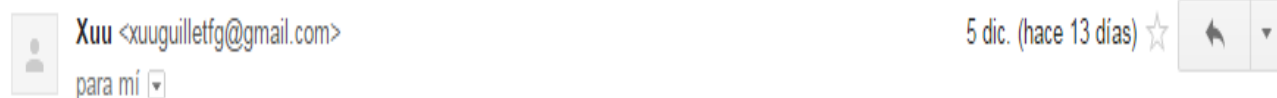
Figura 22 Formulario para cambiar la contraseña

Por último al pinchar en restablecer la contraseña se habrá finalizado con éxito la recuperación de la contraseña.

2.3.16 Activación de la cuenta

Como se ha mencionado anteriormente, puede ocurrir que cuando un usuario se quiera registrar, se encuentre con que la cuenta ha sido creada previamente por el administrador, para poder entrar en este caso se debe activar la cuenta.

Para poderla activar se recibirá un mensaje al correo electrónico. Este mensaje contendrá un enlace en el que se tendrá que pinchar.



Para dar de alta su cuenta pinche en este enlace <http://localhost/tfg5/public/password/confirmacion/xuuguilletfg@gmail.com/profesor/3d9a946c53a9703b5a594fb4776aeb6048e630011f0353a32b60cf72a0e93e22>

Figura 23 Notificación por correo para dar de alta una cuenta

Aparece una vista con un botón llamado:

- **Dar de alta**

Al pinchar en ese botón la cuenta habrá sido dada de alta correctamente y se podrá acceder a la aplicación con esa cuenta.

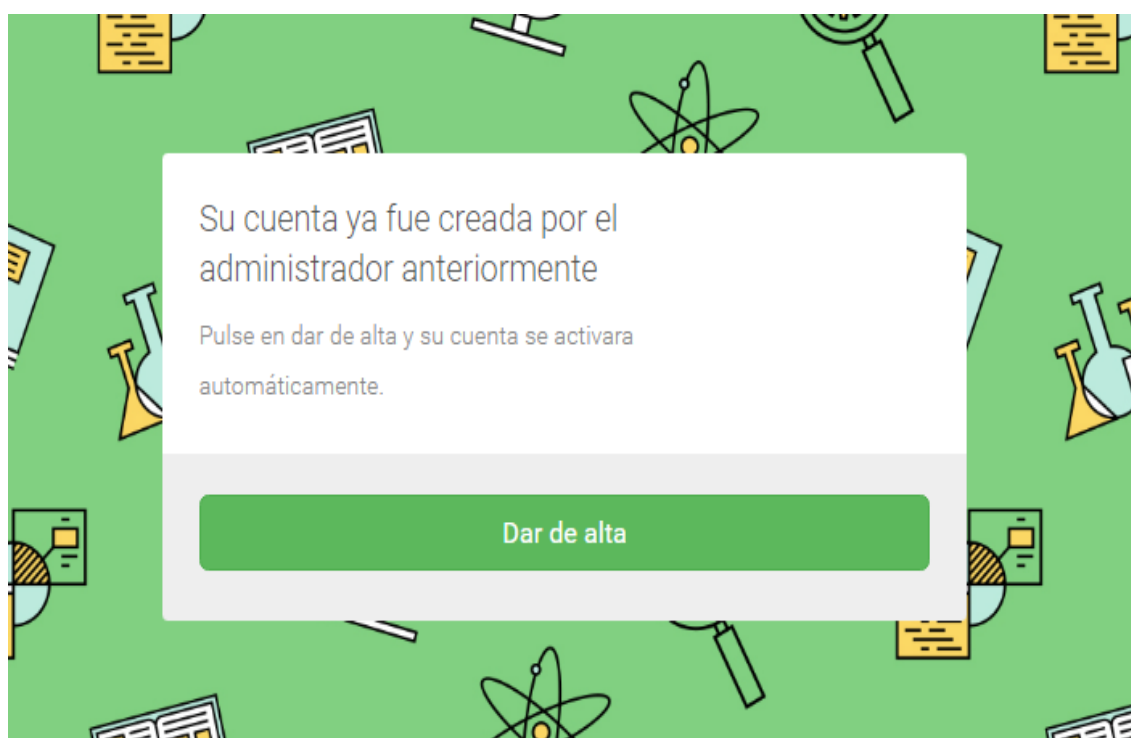


Figura 24 Botón para dar de alta la cuenta

2.3.17 Descargar ficheros de prueba

Para el usuario sería de gran ayuda, que en todos los sitios en los que deba subir un archivo, pudiera descargar antes un fichero que contenga la misma estructura que el que deba subir. Por lo que está implementada esta funcionalidad.

La aplicación, con esta funcionalidad, es más sencilla para el usuario.

2.3.18 Estadística con los datos obtenidos del boletín de notas y del Moodle

Esta funcionalidad aparece cuando el profesor o el evaluador han subido un fichero con el boletín de notas y con los Logs y después acceden a la visualización dinámica del resultado general.

La estadística descriptiva es la rama de las matemáticas que recolecta, presenta y caracteriza un conjunto de datos (por ejemplo, edad de una población, altura de los estudiantes de una escuela, temperatura en los meses de verano, etc.)

Esta funcionalidad que tiene la aplicación obtiene la nota media, porcentaje de los alumnos presentados, porcentaje de los alumnos aprobados del examen final, la nota media, el total de los alumnos y el porcentaje de los alumnos aprobados de una asignatura.

Resultados académicos (examen final)			
Curso	Nota media	Porcentaje de alumnos presentados	Porcentaje de alumnos aprobados
14-70632	2.4	70%	20%

Resultados académicos (nota final)			
Curso	Nota media	Total de estudiante	Porcentaje de alumnos aprobados de la asignatura
14-70632	4	10	60%

Figura 25 Resultado Académico

Por otro lado también obtiene el Top 5 de los alumnos más participativos en una asignatura y porcentaje de las actividades realizadas en la plataforma Moodle.

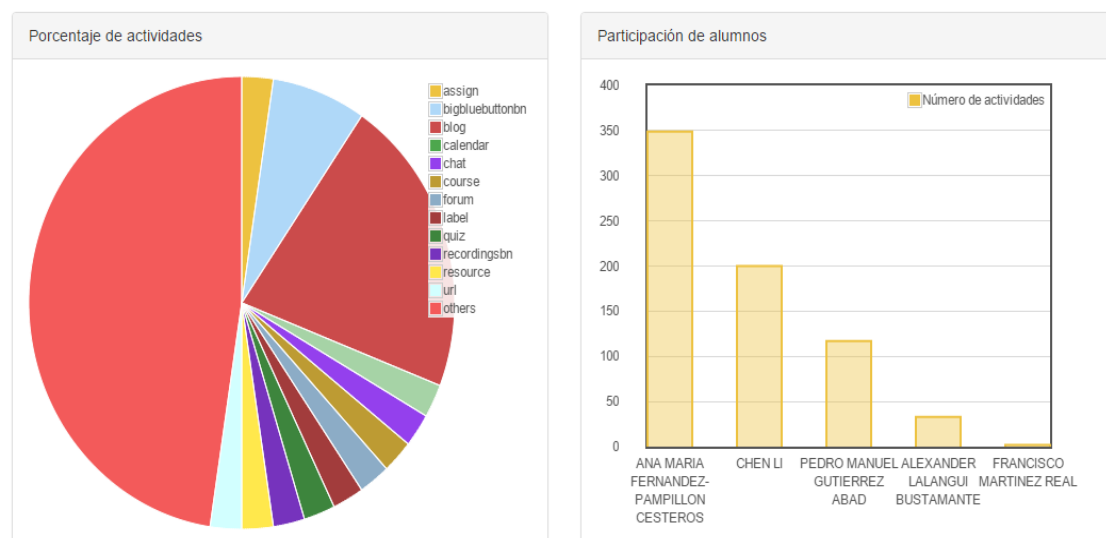


Figura 26 Gráficas del porcentaje de actividades y de participación de alumnos

CAPÍTULO 3: Tecnologías empleadas

Las tecnologías empleadas se pueden clasificar en dos categorías:

1. Cliente

2. Servidor.

1. En la parte del **Cliente** se ha utilizado HTML y CSS. Además se ha empleado un framework llamado [Bootstrap \[5\]](#), que permite al cliente poder navegar a través del contenido de forma fácil e intuitiva.
2. En la parte del **servidor**, el servidor web es [MySQL \[6\]](#) y el lenguaje de programación scripts PHP y se utiliza el framework [Laravel \[7\]](#).

3.1 Laravel: Framework PHP

Para agilizar el trabajo y hacer la aplicación más eficiente se ha utilizado el framework Laravel. Es un framework de código abierto para PHP que implementa el patrón MVC.

Existen otros frameworks similares como [Symfony \[8\]](#) o [CodeIgniter \[9\]](#), pero Laravel tiene una gran cantidad de documentación tanto en su página web como en foros externos, además la información aparece de forma muy clara y por eso esta aplicación utiliza este framework.

3.2 XAMPP: Servidor web

Esta aplicación utiliza XAMPP ya que da muy buenos resultados de rendimiento en pequeñas y medianas aplicaciones. Existen otros como [NGINX \[10\]](#).

3. 3 MySQL: Base de datos

El proyecto tiene que tener una base de datos necesariamente, donde poder guardar toda la información.

Hay dos tipos de base de datos que se pueden utilizar en esta aplicación: SQL o no SQL.

El tipo de base no SQL como [MongoDB \[11\]](#), funciona muy rápido cuando hay mucha información y además tiene una gran escalabilidad.

Esta aplicación utiliza una base de datos SQL llamada MySQL, debido a que la información que maneja está bien estructurada, esta viene instalada con XAMPP.

3. 4 HTML5, CSS y Bootstrap

Para el diseño de la página web se ha utilizado HTML5 con hojas de estilo CSS y ayudado del framework Bootstrap. Este facilita mucho el trabajo y además permite un diseño sencillo y bonito.

CAPÍTULO 4: Arquitectura de la aplicación

Al principio del proyecto el contenido era estático, es decir, no había un modelo de interacción entre las partes. Esto cambió al programar en PHP.

4.1 Aplicación Cliente - Servidor

Para poder entender la aplicación es necesario saber cómo funciona la arquitectura cliente-servidor.

El cliente es aquella persona que se conecta a un navegador web ya sea mediante un móvil, ordenador o tableta y al interactuar con la parte gráfica de la aplicación realiza peticiones a otro programa que es el servidor. Este se encarga de procesar esa petición y devolver la información correspondiente mediante un archivo HTML.

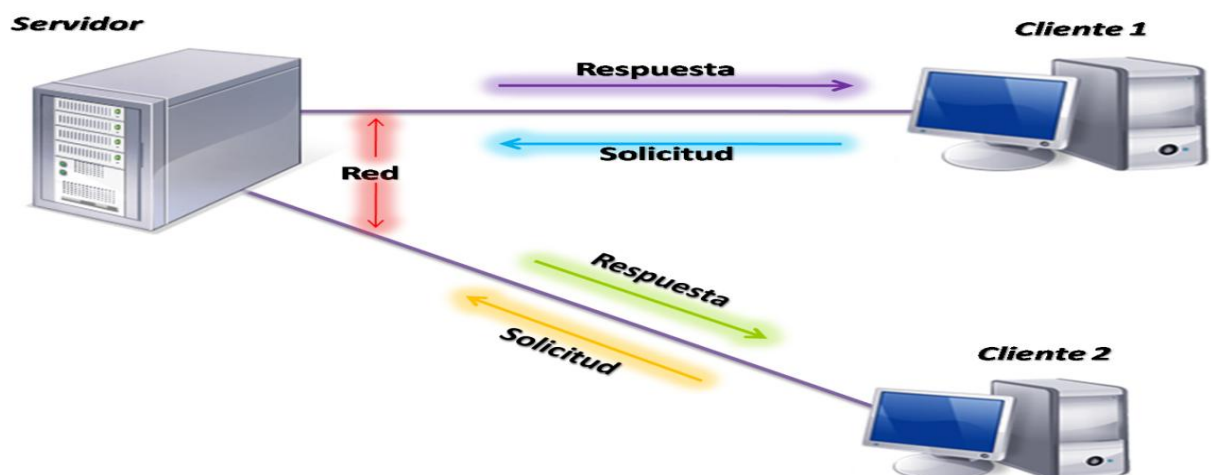


Figura 27 Ejemplo de Cliente-Servidor

A continuación hay un ejemplo detallado de cómo funcionaría en la aplicación.

Un usuario, en este caso el cliente, se conecta con un ordenador a la aplicación mediante el navegador [Google Chrome](#). Primero realizará una petición al servidor accediendo a la [URL](#) específica de la web, por ejemplo <http://localhost/login>.

En el servidor sucederá lo siguiente:

1. En primer lugar XAMPP procesará esta petición y le redirigirá a la aplicación.
2. En segundo lugar, en la aplicación, Laravel contiene un archivo routes.php que contiene todas las rutas URL y define quien se

encargará de procesar esa petición para que una vez procesada reenvíe un fichero HTML con la respuesta.

A continuación se muestra una parte del archivo routes.php y luego un esquema resumiendo el funcionamiento explicado anteriormente.

```
Route::controller('login', 'LoginController', [
    'getRegistrar' => 'login.registrar',
    'postRegistrado' => 'login.registrado',
    'postStore' => 'login.store',
    'getConfirmacion' => 'login.confirmacion',
    'getIndex' => 'login.index',
]);

Route::controller('evaluador', 'EvaluadorController', [
    'getAsignaturas' => 'evaluador.asignaturas',
    'getEvaluar' => 'evaluador.evaluar.{nombre}.{grupo}.{curso}',
    'getEliminar' => 'evaluador.eliminar.{asignatura}',
    'getEliminardefinitiva' => 'evaluador.eliminardefinitiva.{asignatura}',
    'getSolicitarasig' => 'evaluador.solicitarasig',
    'postSubirimagen' => 'evaluador.subirimagen',
    'getGestionsolicitud' => 'evaluador.gestionsolicitud.{asignatura}',
    'getEvaluadas' => 'evaluador.evaluadas',
    'getNoevaluadas' => 'evaluador.noevaluadas',
]);
```

Figura 28 routers.php

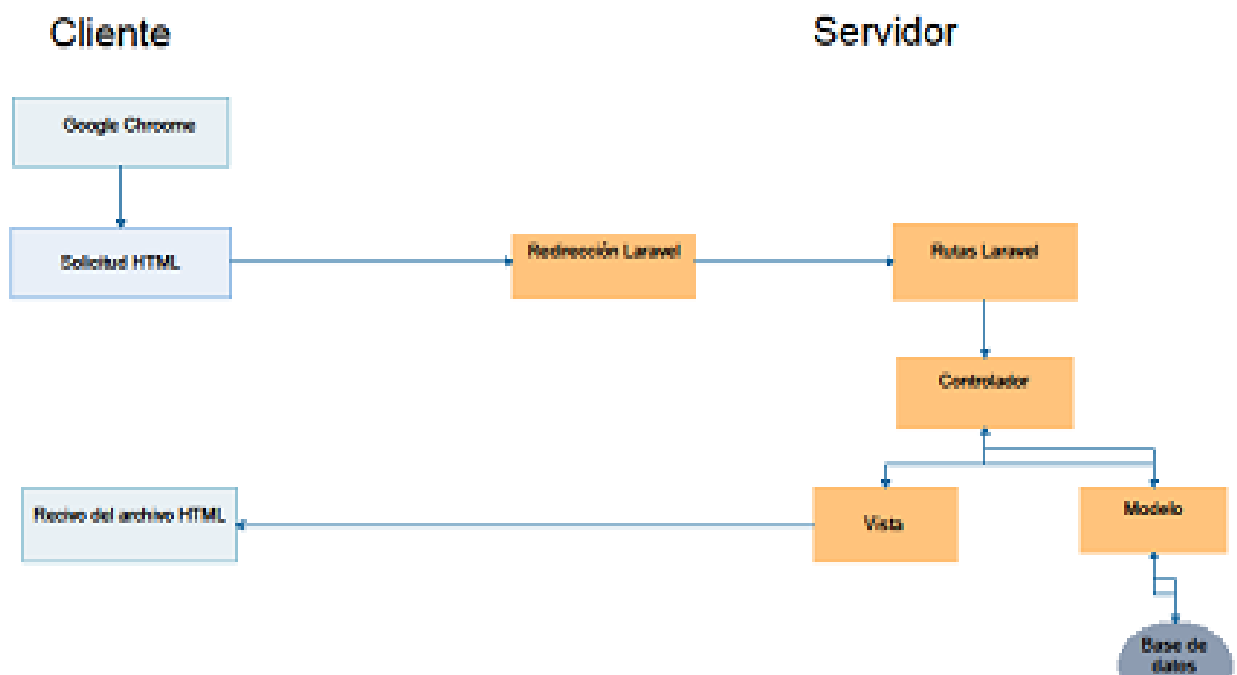


Figura 29 Ejemplo Cliente-Servidor en nuestra aplicación

Hay que destacar que existen principalmente dos formas de comunicar al cliente con el servidor que son mediante los métodos:

1. **GET**
2. **POST**

GET enviara la información mediante la URL y **POST** lo hace en el cuerpo de un mensaje HTTP.

4.2 Estructura Laravel

La estructura que utiliza esta aplicación está basada en el patrón MVC.

A continuación se muestra un diagrama en el que se puede apreciar claramente la estructura.

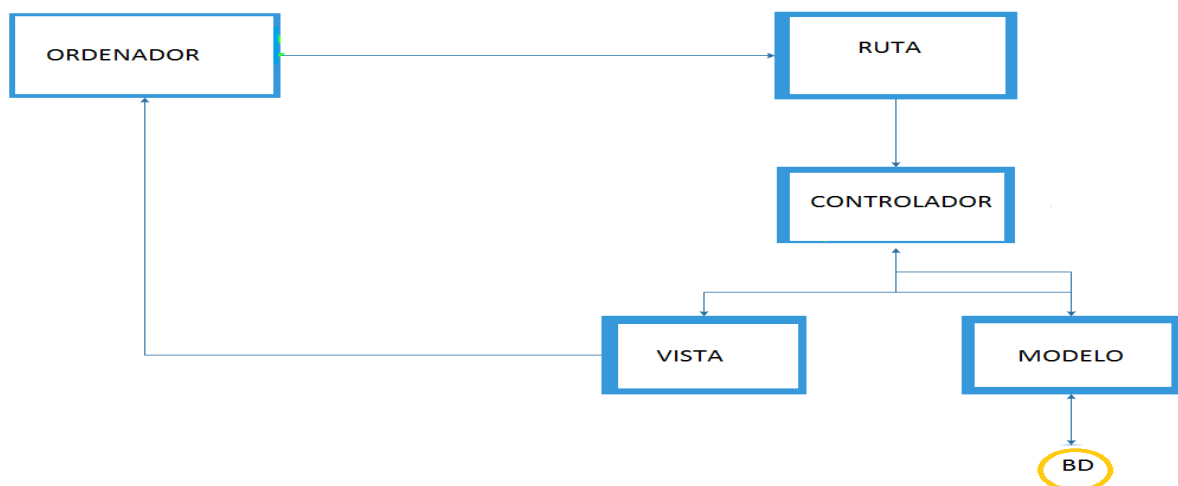


Figura 30 Modelo-Vista-Controlador

La aplicación consta de seis controladores, debido a la función que desempeña cada controlador.

Estos son:

- **Administradorcontroller**
- **Evaluadorcontroller**
- **Logincontroller**
- **Mailcontroller**
- **Profesorcontroller**
- **UsuarioController.**

Para poder crear un controlador hay que ir a la consola y escribir el comando:

php artisan make:controller NombreDelControlador.

CAPÍTULO 5: Modelo de datos

La aplicación tiene que tener necesariamente una base de datos. A continuación se va a explicar en los diferentes apartados como esta implementada.

5.1 Laravel Migration – Creación de tablas

Laravel tiene una funcionalidad que permite crear las tablas en la base de datos.

Para ello hay que crear una clase que extienda de [Migration \[13\]](#) y crear dos métodos llamados:

1. Up

2. Down.

Para explicarlo se va a utilizar de ejemplo la clase CreateUsersTable.

```
class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            // COLUMNS
            $table->increments('id');
            $table->string('email');
            $table->string('rol')->dropColumn(['profesor', 'evaluador', 'administrador']);
            $table->string('password', 20);
            $table->string('nombre');
            $table->string('apellidos');
            $table->string('dni', 9);
            $table->date('fechaNacimiento');
            $table->integer('confirmado')->unsigned()->dropColumn([0, 1, 2]);
            $table->string('imagen');
            $table->rememberToken();
            $table->timestamps();

            // UNIQUE
            $table->unique(array('email', 'rol'));
        });
    }

    /**
     * Reverse the migrations.
     * @return void
     */
    public function down()
    {
        Schema::drop('users');
    }
}
```

Figura 31 Users-Migration

Como muestra la imagen se han creado los dos métodos mencionados anteriormente.

1. En el primero **Up** se define la creación, indicando las columnas que se desean tener en la tabla, como por ejemplo, que haya un campo correo que sea un String.

A continuación se indican las restricciones o referencias. En este ejemplo email y rol tienen que ser únicos.

2. En el segundo método **Down**, se indica el nombre de la tabla para que pueda ser borrada.

Una vez creada estas clases que serán las tablas de la BD, hay que ir a la consola e introducir este comando **php artisan migrate** y automáticamente se crearán las tablas en la base de datos.

5.2 Tablas que tiene nuestra BD

Esta aplicación tiene en total once tablas:

1. **Users**
2. **PasswordResets**
3. **Asignaturas**
4. **Versionesmoodle**
5. **Evaluaciones**
6. **Solicitudes**
7. **Preguntasautomáticas**
8. **Moodles**
9. **Formularios**
10. **Criterios**
11. **Boletin**

A continuación se presenta una breve explicación.

1. **Users**

Esta tabla hace referencia al usuario en la aplicación. Consta de los campos: nombre, apellidos,...etc.

Destacar que el email y el rol son los campos que distinguen a un usuario de otro y por lo tanto en la base de datos estos dos campos deben ser únicos. Por ejemplo, no puede existir dos usuarios con el correo xuuyquille@gmail.com con el rol profesor.

Por razones de eficiencia y implementación email y rol, no son clave primaria, sino que la primary key es un integer auto incremental, debido a que es más rápido buscar por un entero que por dos string. Además Laravel trabaja con un integer incremental para los inserts, deletes, selects y updates.

Por último, mencionar el campo confirmado. Este es un integer que solo puede tomar tres valores:

- **0 el usuario no ha sido confirmado.**
- **1 el usuario ha sido confirmado y puede entrar.**
- **2 el usuario ha sido creado pero no se ha dado de alta.**

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {

        // COLUMNS
        $table->increments('id'); // Integer incremental y primary key
        $table->string('email');
        $table->string('rol')->dropColumn(['profesor', 'evaluador', 'administrador']);
        $table->string('password', 80);
        $table->string('nombre');
        $table->string('apellidos');
        $table->string('dni', 9);
        $table->date('fechaNacimiento');
        $table->integer('confirmado')->unsigned()->dropColumn([0, 1, 2]);
        $table->string('imagen');
        $table->rememberToken();
        $table->timestamps();

        // UNIQUE
        $table->unique(array('email', 'rol'));

    });
}
```

Figura 32 Users-Migration

2. PasswordResets

Esta tabla sirve para poder recuperar la contraseña en caso de que al usuario se le haya olvidado.

Tiene dos campos:

- **email** que hace referencia al correo de un usuario.
- **token** que sirve para poder demostrar la autenticación del usuario en caso de perder la contraseña.

```
public function up()
{
    Schema::create('password_resets', function (Blueprint $table) {
        // COLUMNS
        $table->string('email')->index();
        $table->string('token')->index();
        $table->timestamp('created_at');
    });
}
```

Figura 33 Password Resets_resets-Migration

3. Asignaturas

Esta tabla corresponde a lo que es una asignatura.

Tiene cuatro campos que la diferencia de las demás asignaturas que son:

- **Nombre**
- **Curso**
- **Grupo**
- **Profesor que la imparte.**

Además de los campos anteriores existen otros campos que son:

- **Evaluador, debido a que la asignatura puede tener un evaluador.**
- **Cuatro campos integer**

Dos de los campos integer son:

- **EvaluadaP**
- **EvaluadaE**

Estos campos solo pueden tomar tres valores:

- 0 en caso de que la asignatura no haya sido evaluada.
- 1 si la asignatura ha sido evaluada pero no se ha finalizado de evaluar
- 2 si la asignatura ha sido evaluada.

EvaluadaP controla estas posibles acciones en el profesor y **evaluadaE** en el evaluador.

Por último, otros dos integer que son:

- **BorradaP** (borrada profesor)
- **BorradaE** (borrada evaluador)

Estos campos sólo pueden tomar tres valores:

- 0 la asignatura no ha sido borrada.
- 1 la asignatura ha sido borrada.
- 2 la asignatura ha sido borrada definitivamente.

Se utiliza el método ON DELETE CASCADE para que en el caso de que un profesor se borre también lo haga la asignatura asociada a él.

```
public function up()
{
    Schema::create('asignaturas', function (Blueprint $table) {
        // COLUMNS
        $table->increments('id');
        $table->string('nombre');
        $table->string('curso');
        $table->string('grupo');
        $table->integer('profesor')->unsigned();
        $table->integer('evaluadaP')->unsigned()->dropColumn([0, 1, 2])>default(0);
        $table->integer('evaluador')->unsigned()->nullable()->default(null);
        $table->integer('evaluadaE')->unsigned()->dropColumn([0, 1, 2])>default(0);
        $table->integer('borradaP')->unsigned()->dropColumn([0, 1, 2])>default(0);
        $table->integer('borradaE')->unsigned()->dropColumn([0, 1, 2])>default(0);
        $table->timestamps();

        // UNIQUE
        $table->unique(array('nombre', 'curso', 'grupo', 'profesor'));

        // REFERENCES
        $table->foreign('profesor')->references('id')->on('users')->onDelete('cascade');
        $table->foreign('evaluador')->references('id')->on('users');
    });
}
```

Figura 34 Asignaturas-Migration

4. Versionesmoodle

Como la estructura de los Logs de Moodle puede cambiar con respecto a la versión que se esté utilizando, esta tabla permite definir con que versiones de Moodle se puede trabajar, para poder leer y guardar correctamente los logs.

Para ello esta tabla cuenta con dos campos:

- **Versión**
- **Ruta**

El primero es un string que tiene que ser único, ya que no pueden existir dos versiones con el mismo nombre.

El segundo es un string que almacena la ruta al script correspondiente, se encarga de leer y guardar adecuadamente en la base de datos los logs correspondientes a esa versión de Moodle.

```
public function up()
{
    Schema::create('versionesmoodle', function (Blueprint $table) {
        // COLUMNS
        $table->increments('id');
        $table->string('version');
        $table->string('ruta');
        $table->timestamps();

        // UNIQUE
        $table->unique('version');
    });
}
```

Figura 35 VersionesMoodle-Migration

5. Evaluaciones

Esta tabla se utiliza para guardar el resultado de las evaluaciones. Para ello cuenta con:

- **Un integer que hace referencia al ID de la asignatura.**
- **Un booleano que indica el rol (1 profesor, 0 evaluador).**
- **Un integer con el número de la pregunta.**
- **Un string que se encarga de guardar la puntuación.**
- **Un integer que hacer referencia al a versión del formulario.**

```

public function up()
{
    Schema::create('evaluaciones', function (Blueprint $table) {

        // COLUMNS
        $table->increments('id');
        $table->integer('asignatura')->unsigned();
        $table->boolean('rol');
        $table->integer('apartado');
        $table->integer('num');
        $table->string('puntuacion');
        $table->integer('version_id')->unsigned();
        $table->timestamps();

        // UNIQUE
        $table->unique(array('asignatura', 'rol', 'apartado', 'num'));

        // REFERENCES
        $table->foreign('asignatura')->references('id')->on('asignaturas');
        $table->foreign('version_id')->references('version')->on('formularios');

    });
}

```

Figura 36 Evaluaciones-Migration

6. Solicitudes

Esta tabla es necesaria porque un evaluador puede solicitar asignaturas para evaluar por lo que se deben almacenar estas peticiones.

Consta de dos campos:

- **Asignatura**
- **Evaluador**

El primero hace referencia a una asignatura y el segundo hace referencia a un evaluador.

El campo asignatura y evaluador tiene que ser únicos debido a que un evaluador no puede solicitar dos veces una misma asignatura.

Se utiliza el método ON DELETE CASCADE, para que en el caso de que la asignatura o el evaluador sean borrados, la solicitud también sea borrada en ese momento.

```

public function up()
{
    Schema::create('solicitudes', function (Blueprint $table) {

        // COLUMNS
        $table->increments('id');
        $table->integer('asignatura')->unsigned();
        $table->integer('evaluador')->unsigned();
        $table->timestamps();

        // UNIQUE
        $table->unique(array('asignatura', 'evaluador'));

        // REFERENCES
        $table->foreign('asignatura')->references('id')->on('asignaturas')->onDelete('cascade');
        $table->foreign('evaluador')->references('id')->on('users')->onDelete('cascade');

    });
}

```

Figura 37 Solicitudes-Migration

7. Preguntas automáticas

Esta tabla es necesaria porque si un formulario tiene preguntas automáticas hay que indicar la ruta al script, este es capaz de obtener la valoración de esas preguntas. Por eso, esta tabla tiene dos campos que son:

- **Versión**
- **Ruta**

El primero es un integer que hace referencia a un formulario y el segundo es un string que guarda la ruta al script para obtener la valoración de las preguntas automáticas.

Se utiliza el método ON DELETE CASCADE para que en el caso de que se borre el formulario al que hace referencia, automáticamente se borre la fila.

```

public function up()
{
    Schema::create('preguntasautomaticas', function (Blueprint $table) {

        // COLUMNS
        $table->increments('id');
        $table->integer('version')->unsigned();
        $table->string('ruta');
        $table->timestamps();

        // UNIQUE
        $table->unique('version');

        // REFERENCES
        $table->foreign('version')->references('version')->on('formularios')->onDelete('cascade');

    });
}

```

Figura 38 Preguntasautomaticas-Migration

8. Moodles

Esta tabla se utiliza para guarda la información que contienen los Logs del Moodle en la base de datos.

Tiene un campo llamado user que hace referencia al usuario que ha subido el log.

También tiene un campo llamado asignatura que hace referencia a la asignatura con la que ha subido el log.

El resto de campos guardan la información que tiene un log de Moodle.

Se utiliza el método ON DELETE CASCADE para que en el caso de que el usuario o la asignatura sean borrados, se borre automáticamente la fila que hace referencia a ese usuario o asignatura.

```

public function up()
{
    Schema::create('moodles', function (Blueprint $table) {
        // COLUMNS
        $table->increments('id');
        $table->integer('user')->unsigned();
        $table->integer('asignatura')->unsigned();
        $table->string('curso');
        $table->string('hora');
        $table->string('ip');
        $table->string('usuario');
        $table->string('accion');
        $table->string('informacion');
        $table->timestamps();

        // REFERENCES
        $table->foreign('user')->references('id')->on('users')->onDelete('cascade');
        $table->foreign('asignatura')->references('id')->on('asignaturas')->onDelete('cascade');
    });
}

```

Figura 39 Moodle-Migration

9. Formularios

Esta tabla es para guardar los apartados y preguntas que tiene que tener un formulario. Para ello consta de cuatro campos:

- **Versión**
- **Apartado**
- **Número**
- **Pregunta.**

```

public function up()
{
    Schema::create('formularios', function (Blueprint $table) {
        // COLUMNS
        $table->increments('id');
        $table->integer('version')->unsigned();
        $table->integer('apartado');
        $table->integer('numero');
        $table->string('pregunta');
        $table->timestamps();

        // UNIQUE
        $table->unique(array('version', 'apartado', 'numero'));
    });
}

```

Figura 40 Formulario-Migration

10. Criterios

Esta otra tabla permite guardar ciertos criterios que tiene que tener una evaluación como puede ser el rango de puntuación. Tiene tres campos:

- **NombreCriterio**
- **Descripción**
- **Acción.**

El primero es un string que tiene que ser único, ya que no pueden existir dos criterios que se llamen igual.

El segundo es un String que permite guardar la descripción de ese criterio.

Por último, el tercero es un String permite definir lo que hace ese criterio, por ejemplo, que el rango tiene que ser del 0-2.


```

public function up()
{
    Schema::create('criterios', function (Blueprint $table) {
        // COLUMNS
        $table->increments('id');
        $table->string('nombreCriterio');
        $table->string('descripcion');
        $table->string('accion');
        $table->timestamps();

        // UNIQUE
        $table->unique('nombreCriterio');
    });
}

```

Figura 41 Criterios-Migration

11.Boletín

Esta tabla guarda los datos que contiene el fichero del boletín de notas.
Los campos que tiene esta tabla son los siguientes:

- **User**
- **Asignatura**
- **Nombre**
- **Apellidos**
- **Id_Aux**
- **Id_Usuario**
- **Dni**
- **Correo**
- **Curso**
- **Continua**
- **Nota_final**
- **Examen_final**
- **Total**

Todos estos campos menos el campo “Use” y “Continua”, aparecen en esta tabla, porque el boletín de notas que se obtiene al descargarlo del Moodle, contiene todos estos datos.

El campo “User”, hace referencia al usuario que guarda ese boletín en la base de datos.

El campo “Asignatura”, hace referencia a la asignatura que guarda ese boletín en la base de datos.

Por otra parte en el campo “Continua” los usuarios indican si el alumno supero la evaluación continua o no. Por lo tanto sólo puede tomar dos valores. “sí” en caso de que el alumno pase la evaluación continua y “no” en caso contrario.

Se utiliza el método ON DELETE CASCADE porque si se decide borrar al usuario o a la asignatura, también se borre automáticamente la fila que hace referencia a ese usuario o asignatura.

```
public function up()
{
    Schema::create('boletin', function (Blueprint $table) {
        // COLUMNS
        $table->increments('id');
        $table->integer('user')->unsigned();
        $table->integer('asignatura')->unsigned();
        $table->string('nombre');
        $table->string('apellidos');
        $table->integer('id_aux');
        $table->integer('id_usuario');
        $table->string('dni');
        $table->string('correo');
        $table->string('curso');
        $table->string('continua');
        $table->string('examen_final');
        $table->string('examen_extra');
        $table->string('nota_final');
        $table->timestamps();

        // REFERENCES
        $table->foreign('user')->references('id')->on('users')->onDelete('cascade');
        $table->foreign('asignatura')->references('id')->on('asignaturas')->onDelete('cascade');
    });
}
```

Figura 42 Boletin-Migration

Por último en este apartado se muestra un esquema que resume la estructura de la base de datos.

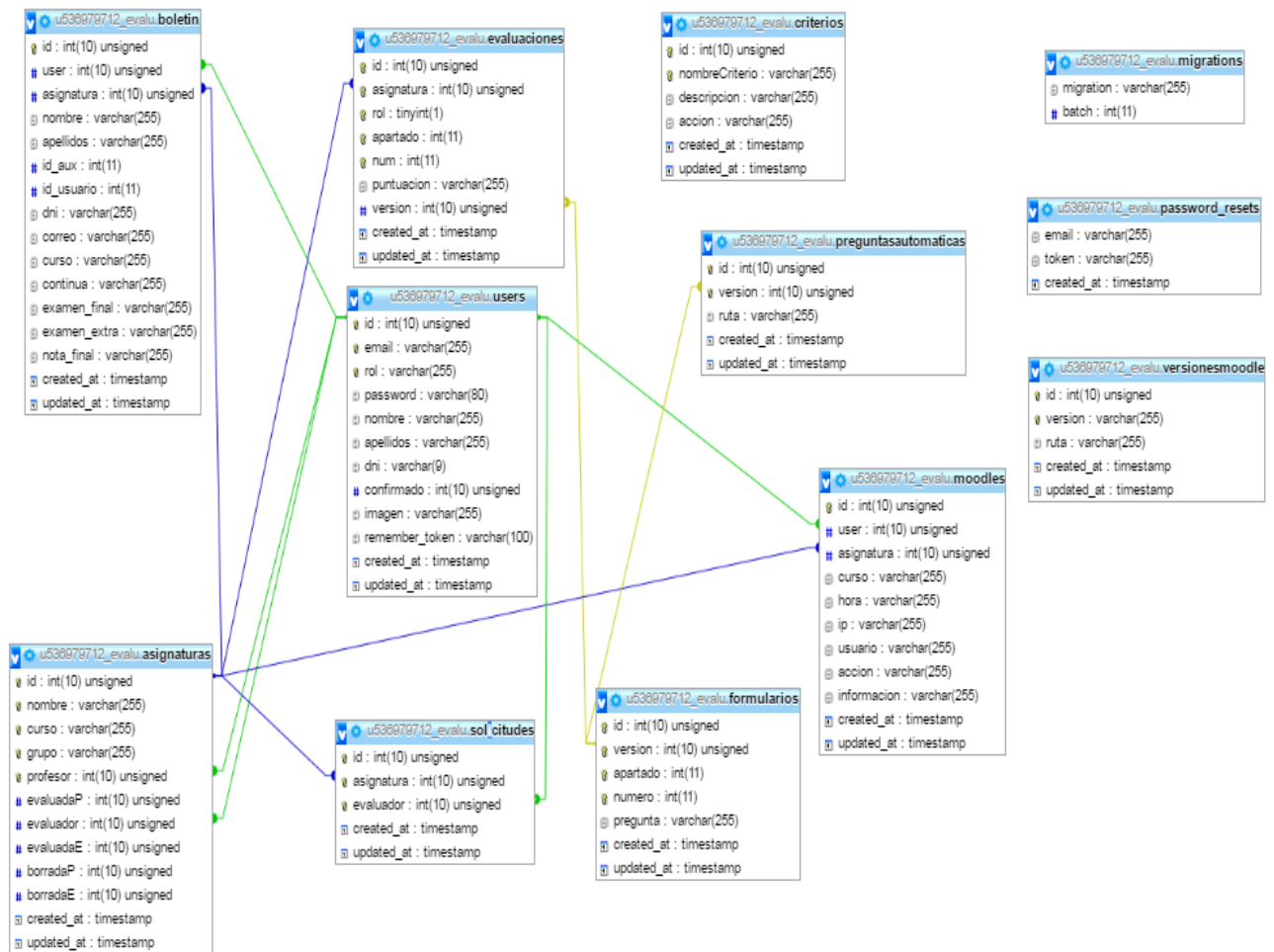


Figura 43 Resumen de la base de datos

5.3 Eloquent ORM - Modelo

Laravel tiene una funcionalidad llamada [Eloquent ORM \[13\]](#) que permite al modelo interactuar con la base de datos.

El principio de esta funcionalidad es que, cada tabla de la base de datos que vaya a usar el modelo tiene que tener una clase que la represente. Esta clase tiene que extender la clase Model, que es la que proporciona toda la funcionalidad de Laravel.

En la siguiente figura aparece un ejemplo de la clase Model User.

```

class User extends Model implements AuthenticatableContract,
                                   AuthorizableContract,
                                   CanResetPasswordContract
{
    use Authenticatable, Authorizable, CanResetPassword;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'users';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = ['email', 'password', 'nombre', 'apellidos', 'dni', 'fechaNacimiento', 'rol', 'confirmado', 'image'];

    /**
     * The attributes excluded from the model's JSON form.
     *
     * @var array
     */
    protected $hidden = ['password', 'remember_token'];
}

```

Figura 44 Model User

En esta clase se especifican atributos como el nombre de la tabla, los nombres de las columnas o atributos ocultos que más tarde servirán para poder recuperar la contraseña.

Laravel permite crear automáticamente estas clases para ello hay que ir a la consola y escribir esta comando **php artisan make:model Nombre-clase-model**.

La aplicación tiene tantas clases model como tablas hay en la base de datos (quitado PasswordReset). En total tenemos diez:

- **Asignatura**
- **Criterios**
- **Evaluación**
- **Formulario**
- **Moodle**
- **Preguntaautomatica,**
- **Solicitud**
- **User**
- **Versionmmodle**
- **Boletín.**

CAPÍTULO 6: Diseño e implementación

6.1 Seguridad

Uno de los temas más tenidos en cuenta de la aplicación es la seguridad.

6.1.1 Protección contra ataques de inyección SQL

La aplicación intenta evitar que el usuario o cualquier persona ajena pueda introducir algún código malicioso. Por eso hay dos capas de seguridad.

Una de ellas se encuentra en la base de datos. Todos los campos de todas las tablas que hay en la base de datos están bien limitados, es decir, si un dato es un String tiene que ser un String,

Si un dato es un entero y es 0 o 1 no podrá tomar otros valores porque está limitado a esos valores.

La segunda capa de seguridad se encuentra a la hora de validar todo tipo de formularios.

Para el tema relacionado con las validaciones, Laravel tiene una funcionalidad llamada [Request \[14\]](#) que permite validar las entradas.

Para crear una de estas clases Request basta con irse a la consola e introducir el siguiente comando **php artisan make:request nombre-clase**.

Una vez creada esta clase tan sólo hay que indicar las restricciones que se deseen.

En la siguiente figura aparece un ejemplo tomado de la clase RegistroRequest.

```

<?php
namespace Evaluate\Http\Requests;
use Evaluate\Http\Requests\Request;
class RegistroRequest extends Request
{
    /**
     * Determine if the user is authorized to make this request.
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     * @return array
     */
    public function rules()
    {
        return [
            'email' => 'required|email',
            'rol' => 'required',
            'nombre' => 'required|string',
            'apellidos' => 'required|string',
            'dni' => 'required|max:9|min:9',
            'fechaNacimiento' => 'required|date',
            'password' => 'required|between:6,12|confirmed',
            'password_confirmation' => 'required|between:6,12',
        ];
    }
}

```

Figura 45 RegistroRequest

6.1.2 Cifrado de las contraseñas

Todas las contraseñas de los usuarios son cifradas. Para ello, Laravel permite encriptar las contraseñas llamando a la función [BCRYPT \[15\]](#), que recibe como parámetro la contraseña que se quiere encriptar.

6.1.3 Autenticación

Para que nadie pueda acceder dentro de la aplicación sino está registrado o alguien que este registrado con un cierto rol no pueda acceder al contenido que tiene otro rol, se utiliza una funcionalidad que tiene Lavarel que se llama [Middleware \[16\]](#).

Esta permite bloquear ciertas rutas.

Como la aplicación consta de 4 controladores (uno para cada rol y otro general para el usuario) existen 4 Middleware:

- **Middleware administrador**
- **Middleware evaluador**
- **Middleware profesor**
- **Middleware usuario**

El **Middleware administrador** se encargará de permitir entrar a las rutas que tiene el administrador, sólo a aquellos usuarios cuyo rol sea administrador. En caso contrario se les dirigirá a la página de Login.

El **Middleware evaluador** se encargará de permitir entrar a las rutas que tiene el evaluador, sólo a aquellos usuarios cuyo rol sea evaluador. En caso contrario se les dirigirá a la página de Login.

El **Middleware profesor** se encargará de permitir entrar a las rutas que tiene el profesor, sólo a aquellos usuarios cuyo rol sea profesor. En caso contrario se les dirigirá a la página de Login.

El **Middleware usuario** se encargará de permitir entrar a las rutas que tiene el usuario en común, como puede ser la vista del perfil sólo a aquellos que sean usuarios de la aplicación. En caso contrario se les dirigirá a la página de Login.

A continuación se muestra como ejemplo el middleware administrador de esta aplicación.

```
<?php

namespace Evaluate\Http\Middleware;

use Closure;
use Session;
use Illuminate\Contracts\Auth\Guard;

/**
 * Evitar que una persona que no tenga el rol de administrador pueda acceder a sus vistas
 */
class Administrador
{
    // Contiene informacion del usuario logeado
    protected $auth;

    /**
     * Constructor por defecto
     */
    public function __construct(Guard $auth){
        $this->auth = $auth;
    }

    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($this->auth->user() == null){
            return redirect()->guest('/login');
        }
        else if($this->auth->user()->rol != "administrador"){
            return redirect()->guest('/login');
        }

        return $next($request);
    }
}
```

Figura 46 Administrador-Middleware

6.2 Envío de correos electrónicos

Para imprimir las notificaciones mediante correos electrónicos, Laravel tiene una funcionalidad llamada [Mail \[17\]](#) que permite el envío de correos electrónicos.

Para poder utilizar esta funcionalidad, primero se ha de crear una cuenta de correo xuugui@autoevaluate.esy.es en [WebMail \[18\]](#), que es el correo que utiliza por defecto el servidor, desde esta cuenta se enviarán los correos electrónicos.

Posteriormente, es necesario configurar los ficheros .env en la carpeta vendor y mail.php en la carpeta config y añadir algunos aspectos como el puerto, el host, la encriptación y la cuenta de correo que se iba a utilizar.

Una vez configurado todo, con tan sólo estas líneas de código, se enviará un correo.

```
$request['mensaje']= 'Su cuenta en Evaluate como profesor ha sido confirmada.';
Mail::send('emails.contact', $request, function($msj) use ($email){
    $msj->subject('Evaluate');
    $msj->to($email);
});
```

Figura 47 Código para enviar un correo

6.3 Recuperación de la contraseña

Para la recuperación de la contraseña y para la activación de una cuenta, Laravel tiene una funcionalidad que viene en modelo User.

Consiste en un campo llamado Token, que tiene la tabla User y que hace referencia a atributo Token de otra tabla llamada Repasword que contiene el email y el Token.

El campo Token no es valor fijo, sino un valor que varía por razones de seguridad para que a si sea más difícil dar con este valor. Lo que hace es enviar un correo electrónico a la dirección que tiene el usuario que quiere recuperar la contraseña.

En el contenido de ese correo, se envía un enlace que contiene la pagina para recuperar la contraseña y en ese mismo enlace se escribe el token que tiene el usuario. Cuando el usuario accede a ese enlace se verifica que el token (aparece en la ruta), coincide con el del usuario y si es así se cambia la contraseña.

6.4 Activación de una cuenta creada anteriormente

Para la activación de una cuenta creada anteriormente, el funcionamiento consiste en enviar un correo electrónico a la dirección que tiene el usuario que quiere activar su cuenta. En el contenido de ese correo se envía un enlace que contiene la página para activar la cuenta y en ese mismo enlace se escribe el token que tiene el usuario.

Cuando el usuario accede a ese enlace se verifica que el token (que aparece en la ruta), coincide con el del usuario y si es así se activa la cuenta.

6.5 Descargar el resultado en PDF

Para poder descargar el resultado en PDF se ha utilizado [DOMPDF \[19\]](#). Este se encarga de convertir el código HTML en un archivo PDF.

Si se quiere utilizar, primero se ha de instalar en Laravel y luego incluir correctamente la ruta donde se encuentra DOMPDF:

```
File::requireOnce('/xampp/htdocs/tfg5/vendor/dompdf/dompdf/dompdf_config.inc.php');
```

Más tarde se debe crear el objeto:

```
$dompdf = new DOMPDF();
```

Y por último con estas líneas de código, se obtiene la descarga del PDF (donde \$html contiene el HTML que aparezca en el PDF):

```
$html = "  
    <html>  
        <body>".  
            $documento."  
        </body>  
    </html>";  
  
$dompdf->load_html($html);  
$dompdf->render();  
  
$dompdf->stream("resultado.pdf");
```

6.6 Separar el código que puede variar con facilidad

Para obtener una aplicación flexible es necesario separar de alguna manera el código que permite leer los logs del Moodle y el código capaz de obtener las preguntas automáticas.

Por eso en la base de datos existe una tabla que, a partir del nombre de la versión de Moodle o del número de formulario que se esté utilizando, se obtiene la ruta al script correspondiente capaz de leer esos logs u obtener las preguntas automáticas.

```
$file = fopen($filename, "r");

$versiones = DB::table('Versionesmoodle')->lists('version');
$version = $versiones[$request['version']];
$rutaversion = DB::table('Versionesmoodle')->where('version',$version)->pluck('ruta');

while(($emapData = fgetcsv($file, 10000, ";")) != FALSE){
    include $rutaversion;
}
```

Figura 48 Código de nuestra aplicación

En el ejemplo anterior del código real del proyecto, se obtiene que versión de Moodle se esté utilizando, una vez que se conoce la versión, se consulta cuál es la ruta al script que es capaz de leer esos Logs y mediante el include \$rutaversion se introduce el código que contiene ese script.

6.7 Organización de las vistas

Todas las vistas se encuentran alojadas en la carpeta TFG/resources/views.

Para una buena el código está dividido en siete carpetas en función de lo que hacen. Estas son:

- **Auth**
- **Emails**
- **Errors**

- **Evaluador**
- **Profesor**
- **Administrador**
- **Usuario**

La carpeta **auth**, contiene todas las vistas relacionadas con el tema de la autenticación, como puede ser login, registro, recuperación de la contraseña,...etc.

La carpeta **emails**, contiene todas las vistas relacionadas con el envío de correos electrónicos.

La carpeta **errors**, contiene la vista de los errores básicos.

La carpeta **evaluador**, contiene todas las vistas con la funcionalidad del usuario con el rol de evaluador, como asignaturas solicitadas,..etc.

La carpeta **profesor**, contiene todas las vistas relacionadas con la funcionalidad del usuario con el rol de profesor, como asignaturas evaluadas por evaluadores,...etc.

La carpeta **administrador**, contiene todas las vistas relacionadas con la funcionalidad del usuario con el rol de administrador, como lista de solicitudes de usuarios,...etc.

Por último, la carpeta **usuario**, contiene todas las vistas que tienen en común los tres tipos de roles, como la vista de configuración del perfil, de cambio de contraseña,...etc.

6.8 Estructuración mediante plantillas

Esta aplicación está estructurada mediante plantillas con el objetivo de poder reutilizar el código, hacerlo más sencillo y estructurado. Para ello se han utilizado plantillas blade.php que soporta Laravel 5.1.

Las carpetas administrador, profesor, evaluador y auth contienen todas ellas un archivo llamado template.php. Este archivo es independiente para cada carpeta y contiene todo el código que tienen en común estas vistas.

Además para separar y reducir más el código de los template.php, se han sacado ciertas partes del código como las hojas de estilo, la barra de navegación y los scripts.

Por otro lado se pueden ver los archivos blade.php, estos contienen la parte que varía de una vista a otra.

Para poder entender esto mejor, a continuación se muestra un ejemplo de cómo funciona.

En la siguiente imagen se puede ver el código de la vista home.blade.php en la carpeta resources\views\profesor.

```
@extends('profesor.template')
@section('home')
<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-12">
            
        </div>
    </div>
</div>
@endsection
```

Figura 49 home.blade.php

En la primera parte aparece un @extend, que dirige al archivo template.blade que se encuentra dentro de la carpeta profesor y cuando en este aparezca la etiqueta home, se introduce las líneas de código que aparecen entre @extend('home') y @endsection, que aparecen en esta imagen.

A continuación, en la siguiente imagen se muestra el código del archivo template.blade.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="Guillermo Martin Duque">
    <title>Profesor</title>
    <!-- style -->
    @include('usuario.style')
</head>
<body>
    <div id="wrapper">
        <!-- Navigation -->
        @include('profesor.navigation')

        <!-- Contenido que varia en casa vista-->
        @yield('home')
    </div>
</body>
@include('usuario.script')
</html>
```

Figura 50 template.blade

En él se observa la etiqueta @yield('home') que es donde se inserta el código visto anteriormente.

Pero además en la cabecera hay un @include('usuario.style'), con esto lo que se consigue es incluir el archivo style de la carpeta resources\views\usuario que contiene todas las cabeceras.

```
<!-- Bootstrap Core CSS -->
{!!Html::style('http://localhost/tfg5/bootstrap/bower_components/bootstrap/dist/css/bootstrap.min.css')!!}

<!-- MetisMenu CSS -->
{!!Html::style('http://localhost/tfg5/bootstrap/bower_components/metisMenu/dist/metisMenu.min.css')!!}

<!-- Timeline CSS -->
{!!Html::style('http://localhost/tfg5/bootstrap/dist/css/timeline.css')!!}

<!-- Custom CSS -->
{!!Html::style('http://localhost/tfg5/bootstrap/dist/css/sb-admin-2.css')!!}

<!-- Morris Charts CSS -->
{!!Html::style('http://localhost/tfg5/bootstrap/bower_components/morrisjs/morris.css')!!}

<!-- Custom Fonts -->
{!!Html::style('http://localhost/tfg5/bootstrap/bower_components/font-awesome/css/font-awesome.min.css')!!}

{!!Html::style('http://localhost/tfg5/bootstrap/img/simbolo.png')!!}
```

Figura 51 style.blade

Se puede observar, que en el cuerpo hay un include('profesor.navigation'), este añade el archivo navigation de la carpeta resources\views\profesor que contiene la barra de navegación del profesor.

Por último hay un include('usuario.script') que añade el archivo script que contiene todos los scripts que tiene esta aplicación y que está situado en la carpeta resources\views\usuario.

Como se ha podido observar todo los códigos son muy limpios, reducidos y además están muy bien estructurados.

6.9 Laravel Collective

Se ha instalado [Laravel Collective \[20\]](#) para reducir y mejorar la estética del código en los formularios y en el HTML en general.

A continuación se muestran dos códigos de cómo era el código antes y el de después de utilizar Laravel Collective.

Antes:

```
<link rel="stylesheet"
href="http://fonts.googleapis.com/css?family=Roboto:400,100,300,500">
<link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">

<div class="form-group">
  <p> Seleccione el tipo de usuario que se desea registrar: </p>
  <input type="radio" name="tipoUsuarioAdministrador"
id="tipoUsuarioAdministrador" value="Administrador">Administrador<br>
  <input type="radio" name="tipoUsuarioProfesor" id="tipoUsuarioProfesor"
value="Profesor">Profesor<br>
  <input type="radio" name="tipoUsuarioEvaluador" id="tipoUsuarioEvaluador"
value="Evaluador">Evaluador<br>
</div>
```

Después:

```
{!!Form::label('Profesor: ', null, ['class'=>'radio-inline'])!!}
{!!Form::radio('rol', 'profesor', true)!!}
{!!Form::label('Evaluador: ', null, ['class'=>'radio-inline'])!!}
{!!Form::radio('rol', 'evaluador')!!}
{!!Form::label('Administrador: ', null, ['class'=>'radio-inline'])!!}
{!!Form::radio('rol', 'administrador')!!}
```

CAPÍTULO 7: Procesamiento automático de la información

Con la referencia del documento proporcionado por los profesores, se obtuvo un algoritmo para obtener la puntuación de algunas preguntas del formulario de forma automática.

Los valores umbrales para la puntuación de cada criterio están definidos en la siguiente tabla:

Puntuación	Valores Criterio
0 (no tiene)	< 40%
1 (tiene parcialmente)	59%-40%
2 (tiene y es adecuado)	100%-60%
NA (No Aplicable)	El evaluador lo marca como NA
NV (No verificable)	No se puede calcular (faltan datos etc.)

El algoritmo por defecto incluye el resultado las siguientes preguntas:

- **Criterio 47: La participación del estudiante (activa o pasiva) en las diferentes actividades.**
- **Criterio 48: El porcentaje de estudiantes presentados al examen final de la asignatura.**
- **Criterio 49: Porcentaje de estudiantes que supera la evaluación continua.**
- **Criterio 50: El porcentaje de estudiantes que supera el curso.**

A continuación se explicará el algoritmo en detalle:

Criterio 47: La participación del estudiante (activa o pasiva) en las diferentes actividades

- La entrada será un Archivo Log.
- La salida será:

Una puntuación: 0 (no tiene), 1 Sí (tiene parcialmente), 2 Sí (tiene y es adecuado), NV: no verificable

	Actividad en evaluación	Actividad en foro	Actividad de conexión
Peso en la participación del estudiante	50%	30%	20%

- Cálculo de los indicadores:

1. Algoritmo para calcular el indicador de Actividad en evaluación.

- Peso del indicador configurable por el evaluador. Por defecto 0,5
 $\$peso_indicador = 0.5;$
- Calcular el número total de assignment del curso procesando el archivo log
- Calcular el número total de quiz del curso procesando el archivo log
- Calcular el número total de lecciones procesando el archivo log
- Inicializar variable que almacena el valor del indicador para cada estudiante
 $\$ratio_assignment$
- Sacar la lista de estudiante, para cada estudiante hacer:
 - Calcular el número assignment enviados por el estudiante procesando el archivo log
 - Calcular el ratio de assignment
 $\$ratio_a = \$ratio_assignment / \$total_assignment;$
 - Calcular el número quiz realizados por el estudiante procesando el archivo log
 - Calcular el ratio de quiz
 $\$ratio_q = \$ratio_quiz / \$total_quiz;$
 - Calcular el número de lecciones visitadas por el estudiante procesando el archivo log
 - Calcular el ratio de lecciones
 $\$ratio_l = \$ratio_lessons / \$total_leccion;$
 - Calcular el valor del indicador
 $\$actividad_assig = round(((\$ratio_a + \$ratio_q + \$ratio_l) / 3), 2);$
 - Crear un array e insertar el valor actividad_assig para cada estudiante
 $\$arrayassig[\$i] = \$actividad_assig;$
- Obtener la media aritmética del valor total para el indicador de actividad en evaluación, si el valor obtenido es menor de 0,20 entonces se tomará la mediana.

2. Algoritmo para calcular el indicador de Actividad en foro.

- Peso del indicador configurable por el evaluador. Por defecto 0,3
 $\$peso_indicador_foro = 0.3;$
- Peso para los foros leídos configurable por el evaluador. Por defecto 0,5
 $\$peso_leido = 0.5;$
- Peso para las aportaciones a los foros configurable por el evaluador. Por defecto 0,5
 $\$peso_enviado = 0.5;$

-Calcular el número total de mensajes en todos los foros procesando el archivo log
Para cada estudiante hacer:

-Calcular el número de mensajes leídos en todos los foros procesando el archivo log

-Calcular el ratio de mensajes leídos.

$\$ratio_leido = \$total_mensajes_leido / \$total_mensajes;$

-Calcular el número de mensajes enviados en todos los foros procesando el archivo log

-Calcular el ratio de mensajes enviados.

$\$ratio_enviado = \$total_mensajes_enviado / \$total_mensajes;$

-Calcular el valor del indicador

$\$actividad_foro = \$peso_leido * \$ratio_leido + \$peso_enviado * \$ratio_enviado;$

-Crear un array e insertar el valor \$actividad_foro para cada estudiante

$\$arrayforo[\$x] = \$actividad_foro;$

-Obtener la media aritmética del valor total para el indicador de actividad en foro, si el valor obtenido es menor de 0,20 entonces se tomará la mediana.

3. Algoritmo para calcular el indicador de Actividad de conexión.

-Peso del indicador configurable por el evaluador. Por defecto 0,2

$\$peso_indicador_conexion = 0.2;$

-Conexiones mínimas semanales configurable por el evaluador. Por defecto 2

$\$conexion_min = 2;$

-Número de semanas configurable por el evaluador. Por defecto 12

$\$num_semanas = 12;$

Para cada estudiante hacer:

-Calcular el número de conexiones realizadas

-Calcular el ratio de conexión de la cada semana

$\$semana_conexion = \$total_conexion / \$num_semanas;$

-Si el ratio de conexión es mayor que el valor de conexión mínima, el resultado es 1, en caso contrario, calcular el valor proporcionado

if ($\$semana_conexion > \$conexion_min$)

$\$conexion_result = 1;$

else

$\$conexion_result = \$semana_conexion / \$conexion_min;$

-Crear un array e insertar el valor de actividad_conexiones para cada estudiante

$\$arrayconexion[\$y] = \$conexion_result;$

-Obtener la media aritmética del valor total para el indicador de actividad de conexiones, si el valor obtenido es menor de 0,20 entonces se tomará la mediana.

- Calculo de la puntuación.

Algoritmo:

$total_indicadores = indicador_assig * actividad_assig_total +$

indicador_foro*actividad_foro_total+
indicador_conexiones*actividad_conexiones_total

si total_indicadores está entre 1 y 0,60 entonces puntuación = 2
si total_indicadores está entre 0,59 y 0,40 entonces puntuación = 1
si total_indicadores es menor de 0,40 entonces puntuación = 0

Criterio 48: El porcentaje de estudiantes presentados al examen final de la asignatura

- La entrada será el boletín de notas.
- La salida será:
 1. Una puntuación: 0 (no tiene), 1 Sí (tiene parcialmente), 2 Sí (tiene y es adecuado), NV: no verificable
 2. Histogramas de estudiantes presentados respecto total alumnos

Se procesa el archivo boletín de notas buscando el nombre del examen final y calcular el porcentaje de estudiantes presentados.

- si porcentaje está entre 1 y 0,60 entonces puntuación = 2
- si porcentaje está entre 0,59 y 0,40 entonces puntuación = 1
- si porcentaje es menor que 0,40 entonces puntuación = 0

Criterio 49: Porcentaje de estudiantes que supera la evaluación continua

- La entrada será el boletín de notas.
- La salida será: Una puntuación: 0 (no tiene), 1 Sí (tiene parcialmente), 2 Sí (tiene y es adecuado), NV: no verificable

Se procesa el archivo boletín de notas buscando el booleano “continua” para verificar si el alumno ha superado la evaluación continua, calculando el porcentaje de estudiantes superados de la evaluación continua.

- si porcentaje está entre 1 y 0,60 entonces puntuación = 2
- si porcentaje está entre 0,59 y 0,40 entonces puntuación = 1
- si porcentaje es menor que 0,40 entonces puntuación = 0
- no verificable si el boletín de notas no proporciona este dato

Criterio 50: El porcentaje de estudiantes que supera el curso

- La entrada será el boletín de notas.
- La salida será:
 1. Una puntuación: 0 (no tiene), 1 Sí (tiene parcialmente), 2 Sí (tiene y es adecuado), NV: no verificable
 2. Histogramas de estudiantes aprobados, suspensos y no presentados.

Procesar el boletín de notas buscando el nombre de la columna que representa la nota final y calcular el porcentaje de estudiantes superados.

- si porcentaje está entre 1 y 0,60 entonces puntuación = 2
- si porcentaje está entre 0,59 y 0,40 entonces puntuación = 1
- si porcentaje es menor que 0,40 entonces puntuación = 0

Criterio 51: Estadísticas (medidas centrales: media, moda, mediana, varianza,...)

En este criterio se comprobará si los alumnos (la media o la mediana) han aprobado la evaluación continua (puntuación criterio 49 = 2) y también el examen final (criterio 50 = 2). Si es así se puntúa con la máxima (2). Pero si ocurre que la media o mediana de alumnos tienen una puntuación regular en la superación del examen final (criterio 50 = 1) el aprovechamiento no ha sido tan bueno, con independencia de la evaluación continua (pero siempre aprobándola es decir con puntuaciones de 1 ó 2), entonces se puntúa con un 1. En otros casos (la media de alumnos tiene un 0 en la evaluación continua y/o en el examen final) se considera que el aprovechamiento ha sido malo (puntuación criterio 51 = 0)

Si no existe evaluación continua (PuntuacionCriterio49=NA) (PuntuacionCriterio50 = 2) entonces asignar 2.

Si (PuntuacionCriterio49 = 2) y (PuntuacionCriterio50 = 2) entonces Puntuacióncriterio51 = 2

Si ((PuntuacionCriterio49 <> 0) and (PuntuacionCriterio50 = 1) entonces Puntuacióncriterio51 = 1

En otros casos Puntuacioncriterio51=0

Mediante las consultas de queries de Laravel desde las tablas de Moodle y Boletín combinados con unas operaciones matemáticas, cuando el usuario sube el log de moodle y el boletín de notas con el formato correspondiente podrá tener el resultado de estas preguntas de forma automática.

CAPÍTULO 8: Conclusiones y trabajo futuro

CONCLUSIONES

En este proyecto hemos utilizado diferentes tipos de tecnologías y por primera vez hemos trabajado con un Framework. Esto nos permitirá en un futuro trabajar con él o con otro Framework, de forma más rápida y precisa ya que hemos aprendido a leer documentación y entender un código que no era nuestro.

Por otro lado, hemos trabajado en todo momento con especificaciones y datos reales, por lo que hemos creado una aplicación destinada al mundo real y no sólo al académico.

Además hemos trabajado en algunos momentos con grandes cantidades de datos lo que nos ha ayudado a aprender como guardar, analizar y abstraer todos estos datos.

Por último, la complejidad de juntar todas estas tecnologías y la dimensión que tiene nuestra aplicación nos ha permitido conocer cómo funcionan las medianas y grandes aplicaciones que se utilizan a diario.

CONCLUSIONS

In this project we have used different types of technologies and it's the first time that we have used a Framework because we have never used one of these before. Laravel allows us to work with others frameworks faster and in a more accurate way in the future because we have already learned to read documentation and understand people's code

Also, we have worked with real specifications and data. For this reason, we have created an application for the real world and not only academically.

In addition, we have worked with a huge amount of data and all these data has helped us to learn how store, analyze and abstracting this information.

Finally, we have learnt how to use a medium or a big application that we normally use daily thanks to putting all these technologies together by creating a great project.

Trabajo futuro

Nos hemos quedado con ganas de seguir avanzando con nuestra aplicación debido a que existen muchos puntos que todavía se pueden explorar y explotar más.

El más destacado sería aumentar el número de preguntas que podemos obtener automáticamente. Pero incluso podemos ser más ambiciosos y podemos observar que hay muchos datos que se podrían relacionar y se podría aplicar algoritmos de Big Data.

Por eso, este proyecto se podría seguir desarrollando y ampliando, con esta nueva tecnología llamada Big Data, pudiendo obtener increíbles resultados y así poder ayudar más a la evaluación docente

Trabajo realizado por los miembros

En este apartado se especifica el trabajo que ha realizado cada integrante de este proyecto.

La asignación del trabajo al principio era común ya que había que buscar los requisitos y ver cómo iba a ser la aplicación.

Una vez que los requisitos se quedaron claros y cómo iba a ser la aplicación, se empezó a hacer el reparto del trabajo.

Cada integrante tenía una parte que implementar y se le asignaba una fecha para tenerlo finalizado. Si se finalizaba antes de tiempo, se asignaba una nueva tarea y en el caso de no cumplir con la fecha, el compañero si dependía de esa tarea para seguir avanzando, ayudaba a la otra persona a terminar su trabajo.

La proporción de trabajo no ha sido la misma para los dos integrantes del proyecto.

Por eso a continuación se describe lo que ha hecho cada uno.

Guillermo Martín Duque

Mi aportación en este proyecto ha sido muy amplia y variada.

Por una parte me he encargado de dirigir el proyecto.

He buscado que tecnologías íbamos a usar, en este caso PHP con el Framework Laravel y he tenido que comprobar cómo funcionaban estas tecnologías.

Me he dedicado a planificar el tiempo del que disponíamos, para que el proyecto se pudiera acabar en el tiempo adecuado y de forma satisfactoria.

Por otro lado, como he mencionado anteriormente, al principio me dediqué junto con mi compañera a **obtener las funcionalidades y requisitos** que debería tener nuestra aplicación.

Los dos juntos hicimos bocetos en papel de cómo iba a ser el diseño de esta aplicación.

Una vez que tuvimos todos los requisitos me dediqué a desarrollar casi todas las **vistas HTML** que se encuentran en resources/view.

Estas vistas al principio eran un archivo HTML pero al implementar la aplicación pasaron a ser un archivo PHP.

Lleve a cabo todo este paso y aplique la funcionalidad que permite Laravel, **vistas estructuradas en forma de plantilla** y usar **Laravel Collective** para reducir el código y hacerlo más claro.

Por otra parte, he implementado **casi todo el código que tiene la aplicación**.

Primero desarrollando **todo el código** relacionado con el **registro** (registro y acceso).

Después he implementado **toda la funcionalidad** que existe en los tres tipos de usuarios (**profesor, evaluador y administrador**), exceptuando el script que se encarga de obtener la puntuación de algunas preguntas automáticas, las gráficas de porcentaje de actividades, participación de alumnos y la de en forma de araña y las tablas del resultado académico que ha sido realizado por mi compañera.

Para realizar toda esta implementación he diseñado y creado la base de datos utilizando Laravel Migration y Eloquent ORM para permitir al modelo interactuar con la base de datos.

He creado todos los controladores que se encuentran app/Http/Controller con el código que hay en estos archivos.

He realizado todo el tema del enrutado que se encuentra en el archivo router.php en la dirección app/Http.

He implementado la funcionalidad que tiene Laravel llamada Middleware que permite bloquear ciertas rutas. Todo este código se encuentra en App/Http/Middleware.

También he implementado la funcionalidad que tiene Laravel llamada Requests para validar los formularios. Todo este código se encuentra en App/Http/Requests

Por otro lado he implementado otras cosas como DOMPdF para la descarga de archivos en PDF, Mail (funcionalidad de Laravel), para el envío de mensajes, y otras implementaciones que hacen posible el funcionamiento relacionado con el registro, el usuario, el evaluador y el administrador.

Posteriormente cuando finalice con la implementación, me he encargado de que, el código estuviera bien estructurado y organizado en las carpetas correspondientes.

Una vez finalizada casi toda la implementación, mi objetivo fue **subir la aplicación a un servidor**.

Esto me resultó un poco complicado, debido a que anteriormente habíamos trabajado en Localhost con un servidor en Windows y el servidor web, donde hemos alojado nuestra aplicación, funciona en Linux, por lo que he tenido muchos problemas con el tema de las mayúsculas y minúsculas.

He tenido un problema con la funcionalidad que tiene nuestra aplicación para enviar correos, debido a que al principio utilizábamos una cuenta de GMAIL para el envío de correos, pero nuestro servidor tenía los puertos SMTP que utiliza GMAIL cerrados (puertos 587 y 465), por lo que decidí crear una cuenta de WebMail ya que nuestro servidor tiene asociado este correo.

Una vez que cree una cuenta de correo en WebMail utilicé el puerto SMTP 2525 ya que estaba habilitado y WebMail puede funcionar con este puerto. Pero tampoco funcionaba ya que consideraba la aplicación de Laravel como insegura.

Después de intentar que funcionase de todas las maneras posibles, decidimos crearnos una cuenta Premium en nuestro servidor [Hostinger \[21\]](#) ya que si pagábamos nos habilitaban los puertos SMTP que utiliza GMAIL.

Una vez que nos creamos la cuenta Premium se solucionó el error y mantuvimos el correo con WebMail.

Por último sobre mi aportación en este proyecto, cabe destacar que me he dedicado a **escribir los manuales de uso e instalación** y a **redactar** casi toda esta **memoria**.

Yuan Xu Jiang

La parte del estudio en la que se centró mi radio de acción principalmente en este proyecto fue la basada en el estudio sobre la evaluación automática.

El algoritmo que se extrae a partir de la evaluación lleva un proceso conciso y concienzudo sobre las variables extraídas en tablas en formato excel donde se incluye toda la información obtenida de los logs de moodle.

A continuación una revisión exhaustiva y la interpretación de dichos resultados mostraron la información que muestran y la cual se pudo extraer a partir de ellos.

Posteriormente y gracias a la inestimable ayuda de la codirectora aportándonos la información necesaria y fundamental con el documento “Proyecto Calidad Prodesorado_v2”. obtuvimos las consultas queries estudiando la documentación del Generador de consultas de Laravel.

Los resultados obtenidos se plasmaron en una gráfica la cual indicó de una forma clara y directa las conclusiones consiguiendo poder interpretarlos de una forma prácticamente inmediata.

El almacenamiento de la información cargada a partir de la evaluación se generó de la siguiente manera: se creó una tabla en base de datos Moodle y otra tabla Boletin.

A continuación se efectuamos las pruebas para garantizar si las consultas son correctas.

Adicionalmente, para los administradores que puedan tener una visión general sobre los resultados de las evaluaciones, mediante los marcos de gráficos de javascript con unos cálculos matemáticos conseguimos mostrar los resultados en diferentes aspectos del formulario en un diagrama de araña.

Es importante indicar la importancia de la colaboración según comenzó el proyecto de todos los miembros del equipo donde se definieron las funcionalidades y los requisitos de la aplicación, desarrollando los casos de uso y los bocetos en diferentes diseños de borradores hasta dar con el diseño final de nuestra aplicación.

He de indicar mi notoria participación en la creación de los diseños HTML.

Por otro lado, cuidé el diseño de la aplicación mediante la utilización de la plantilla Bootstrap para conseguir representar la información y su contenido de una forma sencilla y clara, sin saturar la vista del usuario con información innecesaria y confusa de forma que al final obtuviéramos a una aplicación visualmente simple pero a su vez potente al no perder información vital para el proyecto.

El diseño sobre la unificación del color de la interfaz, iconografía utilizada, pequeños ajustes y en general, la interfaz tanto funcional como gráfica han permitido una aplicación muy fácil de usar.

Para último, completar, revisar y redactar la presente memoria.

Bibliografía

- [1] Moodle. Aplicación web educativa que contiene un sistema de gestión de cursos. [Online]. <https://moodle.org/?lang=es>
- [2] Logs del Moodle. Informes de actividad en Moodle. [Online]. <https://docs.moodle.org/23/en/Logs>
- [3] EVALUATE. Página web de nuestra aplicación. [Online]. <http://autoevaluate.esy.es/>
- [4] Datatables. Plugin para la biblioteca jQuery Javascript. [Online]. <https://datatables.net/>
- [5] Bootstrap. Librería de diseño de aplicaciones web. [Online]. <http://getbootstrap.com/>
- [6] MySQL. Sistema de gestión de base de datos relacional. [Online]. <https://www.mysql.com/>
- [7] Laravel. Framework de código abierto para desarrollar aplicaciones y servicios web con PHP. [Online]. <https://laravel.com/>
- [8] Symphony. Framework de desarrollo en el lado del servidor. [Online]. <https://symfony.com/>
- [9] CodeIgniter. Framework de desarrollo en el lado del servidor. [Online]. <http://www.codeigniter.com/>
- [10] NGINX. Servidor web. [Online]. <http://nginx.org/>
- [11] MongoDB. Base de datos no relacional. [Online]. <https://www.mongodb.org/>
- [12] Migration. Funcionalidad que tiene Laravel. [Online]. <https://laravel.com/docs/5.1/migrations>
- [13] ELOQUENT ORM. Funcionalidad que tiene Laravel. [Online]. <https://laravel.com/docs/5.1/eloquent>
- [14] Request. Funcionalidad de Laravel para validar los formularios. [Online]. <https://laravel.com/docs/5.0/validation#form-request-validation>
- [15] BCrypt. Funcionalidad que tiene Laravel para encriptar contraseñas. [Online]. <https://laravel.com/docs/5.0/hashing>
- [16] Middleware. Funcionalidad que tiene Laravel. [Online]. <https://laravel.com/docs/master/middleware>
- [17] Mail. Funcionalidad que tiene Laravel. [Online]. <https://laravel.com/docs/5.1/mail>
- [18] WebMail. Correo electrónico de nuestra aplicación. [Online]. <http://webmail.hostinger.es>
- [19] DOMPDF. Convertidor de código HTML a un archivo PDF. [Online]. <https://dompdf.github.io/>

[20] Laravel Collective. Funcionalidad que tiene Laravel. [Online].
<https://laravelcollective.com/>

[21] Hostinger. Servidor donde está alojado nuestra aplicación. [Online].
<http://www.hostinger.es/>

Anexo 1: Guía de instalación en el servidor

Introducción

Este manual de instalación va destinado a aquellas personas que desean dejar funcionando la aplicación en el servidor, cuando la base de datos está vacía, es decir cuando en la base de datos sólo existe un usuario con rol de administrador.

Destacar que este manual de instalación deja la aplicación funcionando con el formulario que contiene las preguntas automáticas.

Acceso a la aplicación

Lo primero que hay que hacer es abrir un navegador de internet (preferiblemente Google Chrome) e introducir en la URL la siguiente dirección:

- <http://autoevaluate.esy.es/>

Una vez que se ha accedido a la página web se debe entrar como **Administrador** con la siguiente cuenta:

- Dirección de correo: **xuuguilletfg@gmail.com**
- Contraseña: **admintfg**
- Rol: **Administrador**

Configuración

Después de entrar en la aplicación como Administrador hay que seguir los siguientes pasos para que la aplicación funcione:

1. **Subir un formulario.**
2. **Subir criterios de evaluación.**
3. **Crear la versión Moodle.**

1. Subir un formulario

Para poder subir un formulario hay que ir al menú lateral izquierdo y hacer click en:

Configuración y después en

Formulario

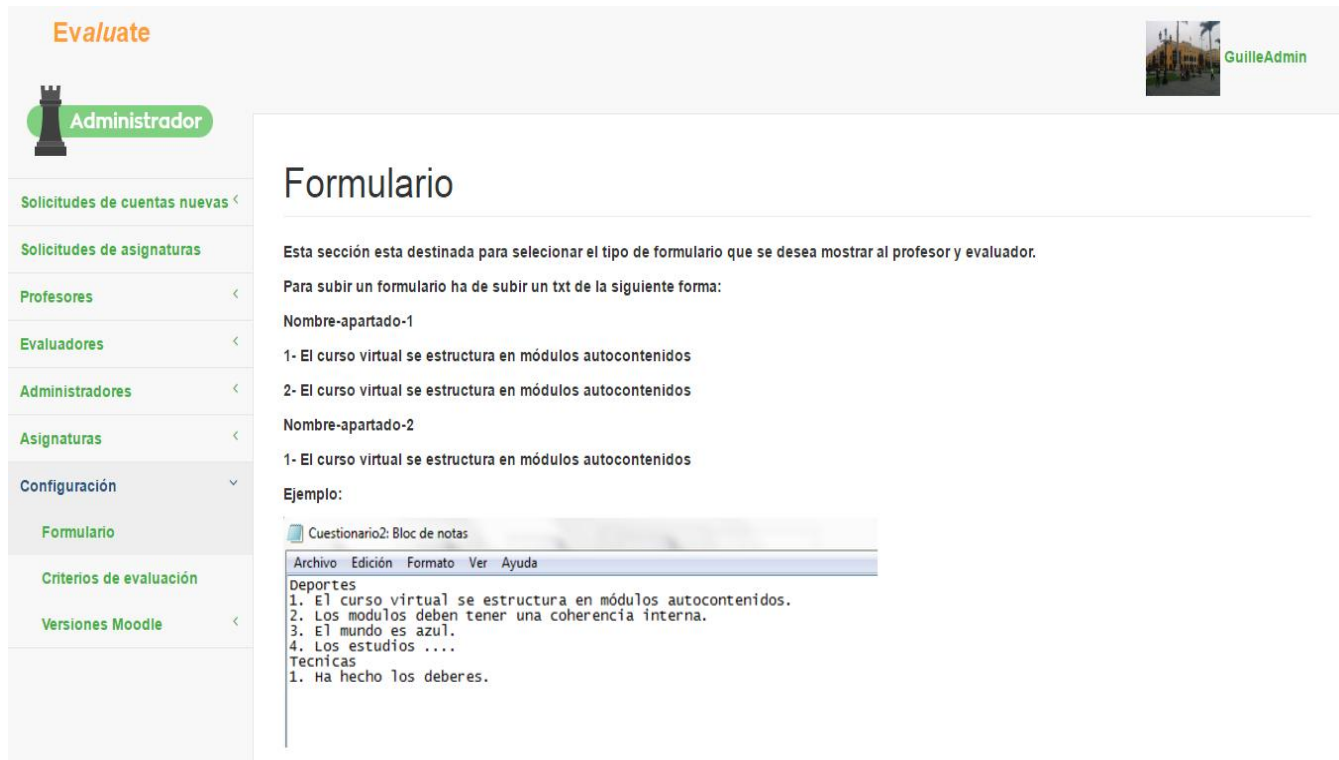


Figura 52 Vista donde se sube el formulario.

Si aparece la vista de la figura anterior se está en el sitio correcto.

A continuación hay que subir un archivo txt con el formulario. Para ello es necesario descargar el fichero txt con formulario pinchando en el botón:

- **Descargar fichero de pruebas.**

Una vez descargado el txt hay que pinchar en el botón:

- **seleccionar archivo**

Se selecciona el fichero txt que se ha descargado anteriormente.

Cuando se haya completado todos estos pasos hay que indicar el archivo que contiene los algoritmos para calcular las preguntas automáticas.

Para ello se debe introducir lo siguiente:

Ruta al script que contiene las preguntas automáticas:
/scripts/preguntasAutomaticas.php

Por último, pinchar en el botón

- **Subir.**

2. Subir criterios de evaluación

Para poder subir los criterios de evaluación hay que ir al menú lateral izquierdo que aparece en la aplicación y hacer click en:

Configuración y después en

Criterios de evaluación

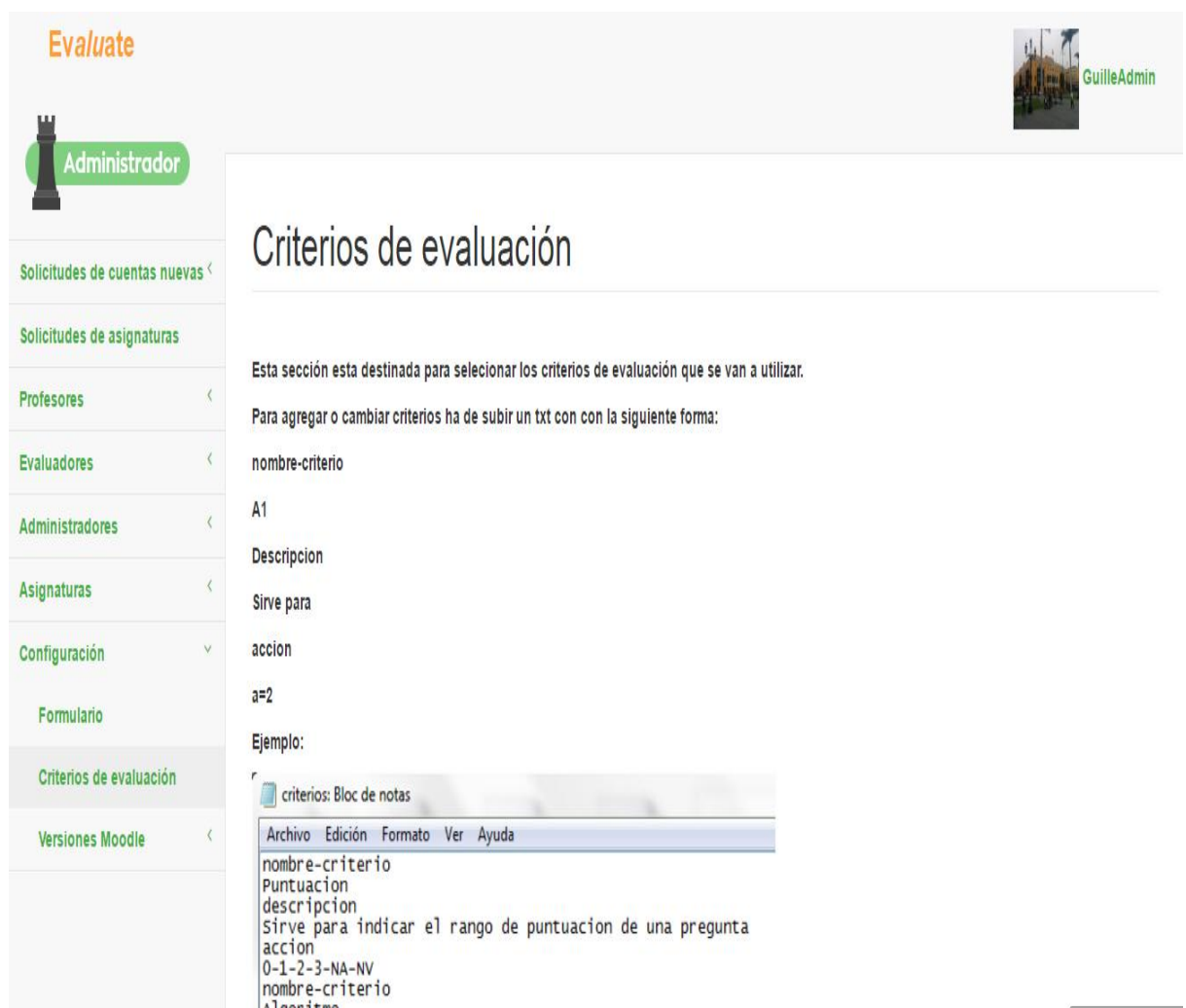


Figura 53 Vista donde subir los criterios de evaluación.

Después hay que pinchar en:

- **Descargar fichero de prueba**

Una vez descargado el txt pinchar en:

- **seleccionar archivo**

Se selecciona el fichero txt descargado anteriormente.

Por último, pinchar en el botón

- **Subir**

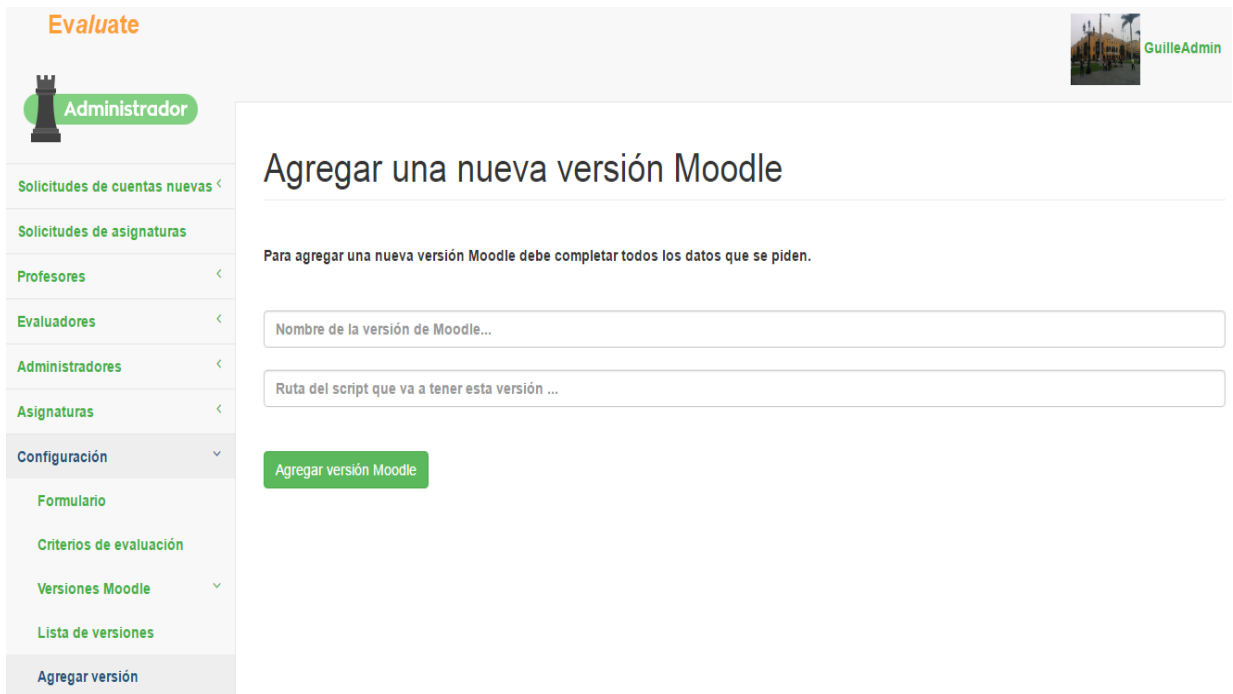
3. Crear la versión Moodle

Para poder crear la versión Moodle hay que ir al menú lateral izquierdo que aparece en la aplicación y hacer click en:

Configuración después en

“Versiones Moodle” y por último en

Agregar versión



The screenshot shows the Moodle application interface. On the left is a sidebar menu with the following items: 'Administrador' (highlighted with a green background), 'Solicitudes de cuentas nuevas', 'Solicitudes de asignaturas', 'Profesores', 'Evaluadores', 'Administradores', 'Asignaturas', 'Configuración' (expanded), 'Formulario', 'Criterios de evaluación', 'Versiones Moodle' (expanded), 'Lista de versiones', and 'Agregar versión'. The main content area is titled 'Agregar una nueva versión Moodle'. Below the title, it says 'Para agregar una nueva versión Moodle debe completar todos los datos que se piden.' There are two input fields: 'Nombre de la versión de Moodle...' and 'Ruta del script que va a tener esta versión ...'. At the bottom of the form is a green button labeled 'Agregar versión Moodle'.

Figura 54 Página donde se agregan las versiones Moodle

Si aparece la vista de la figura anterior se está en el sitio correcto.

Después hay que introducir los siguientes datos:

- Nombre de la versión Moodle: **usted tiene elección en este campo.**
- Ruta al script que va a tener esta versión: **/scripts/VersionMoodle27.php**

Por último, pinchar en el botón

- **Agregar versión Moodle**

Una vez completado estos 3 pasos la aplicación estará instalada correctamente.

Anexo 2: Guía de instalación en Localhost

Para instalar la aplicación en localhost es necesario instalar dos cosas principalmente:

- **XAMPP**
- **Laravel**

INSTALACIÓN DE XAMPP

1. Hay que ir a la página de XAMPP:

<https://www.apachefriends.org/es/index.html>

2. En la página anterior hay que descargar XAMPP para Windows.
3. Cuando se descargue se verá que la descarga es un archivo .exe.
4. Abrir este archivo.

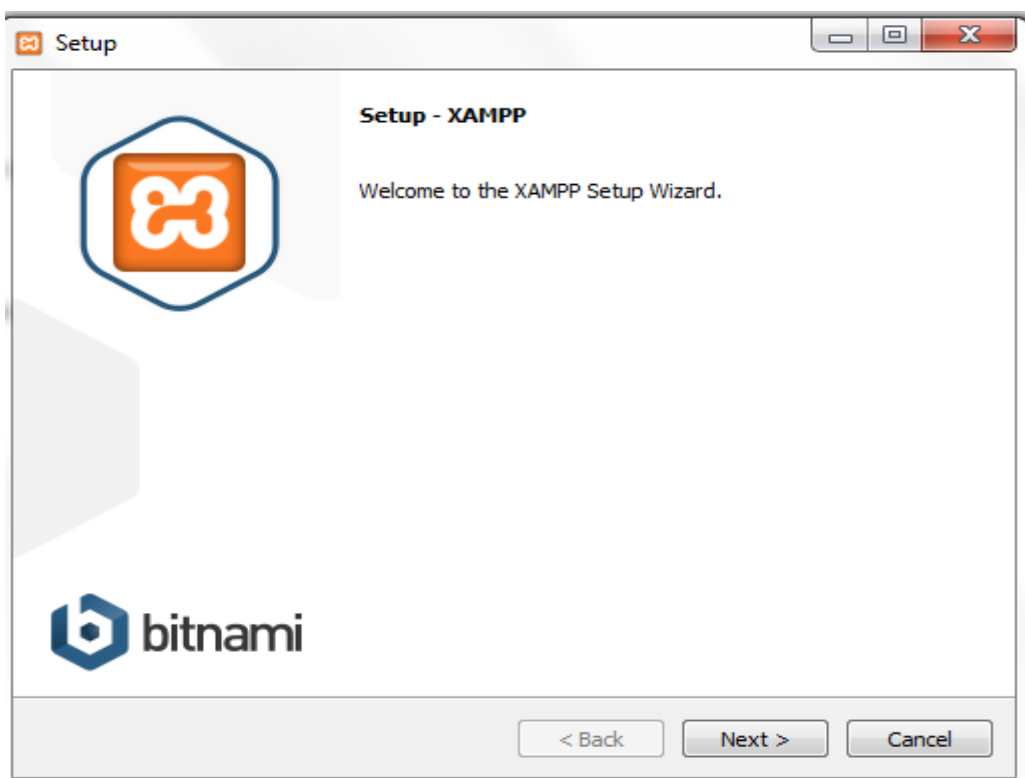


Figura 55 Instalar XAMPP

5. En la figura anterior se pulsa el botón de “Next”.

6. Aparece una nueva pantalla hay que volver a pulsar en “Next”.

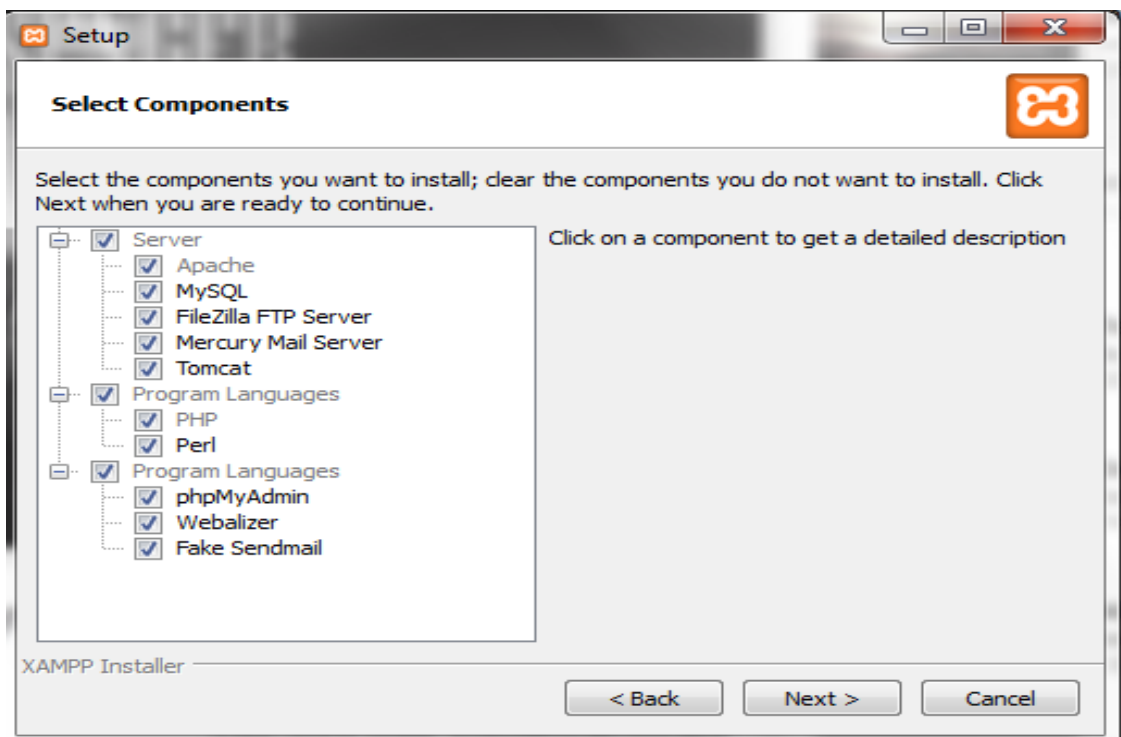


Figura 56 Instalación de XAMPP 2

7. En la nueva pantalla que aparece a continuación indicar que se quiere instalar XAMPP en la ruta: C:\xampp

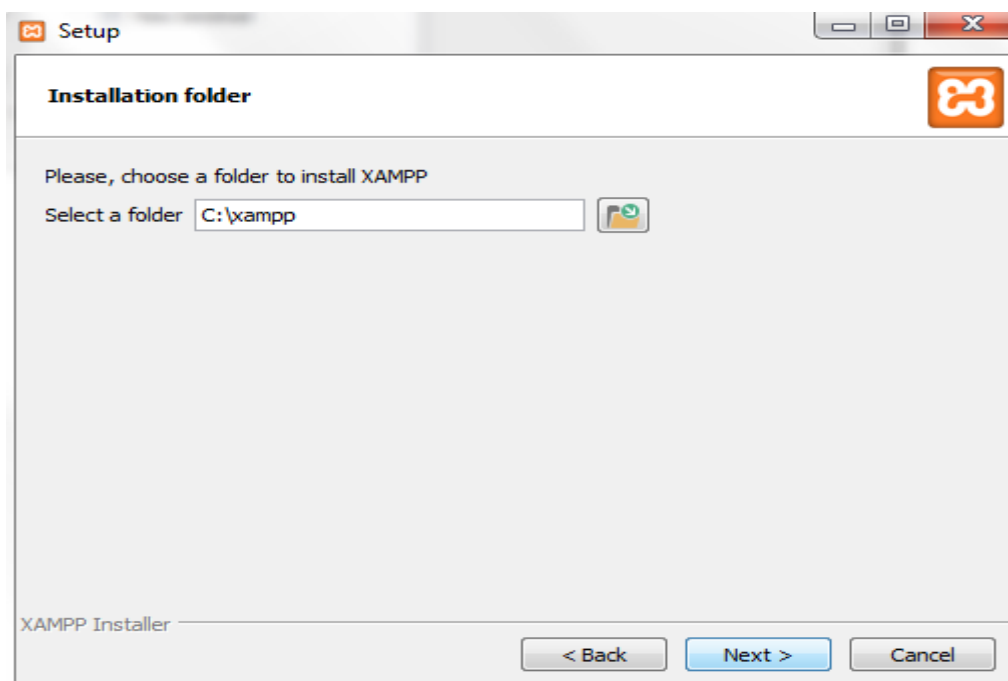


Figura 57 Instalación XAMPP: indicar la ruta

8. A continuación pulsar en “Next” y finaliza la instalación de XAMPP.

INSTALACIÓN DE LARAVEL

1. Lo primero que hay que hacer para instalar LARAVEL es instalar **COMPOSER**.
2. Para instalar COMPOSER hay que ir a la siguiente página:
 - <https://getcomposer.org/download/>
3. En la página anterior descargar COMPOSER haciendo click en **Composer-Setup.exe**.
4. Una vez que la descarga finalice, ejecutar el instalador y haz click en: "Next".

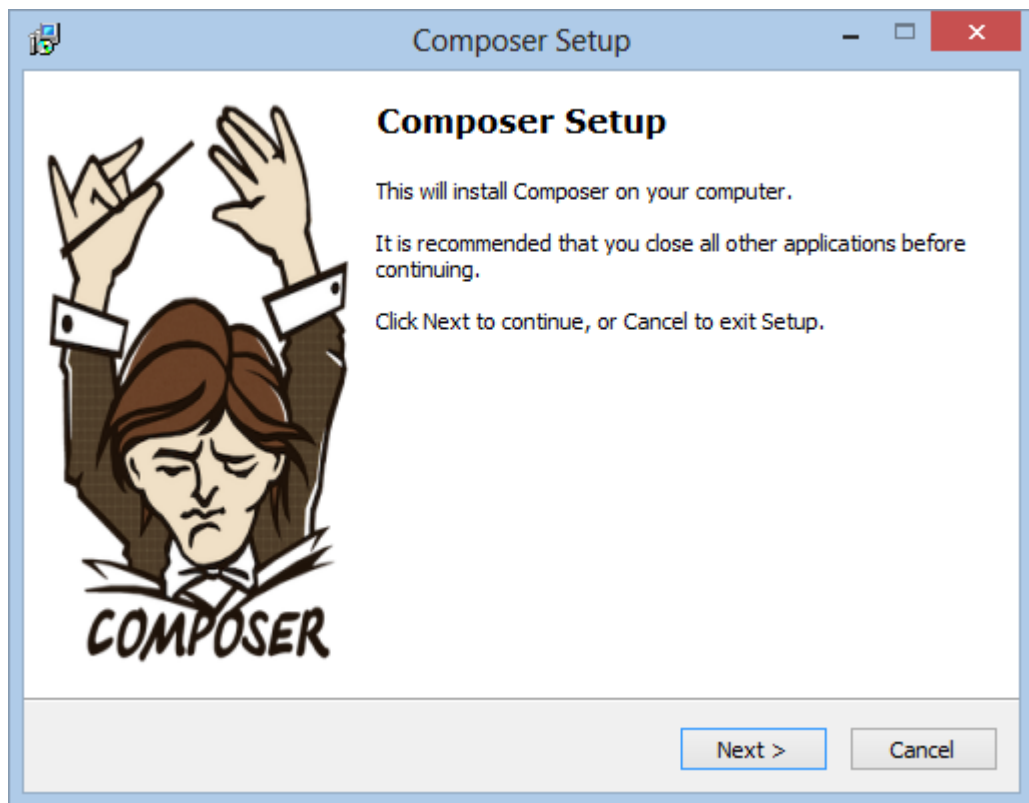


Figura 58 Instalar COMPOSER

5. Si se quiere administrar los proyectos mediante el Explorador de Windows, se puede seleccionar la opción "**Install Shell Menus**", aunque lo recomendable es usar la línea de comandos.

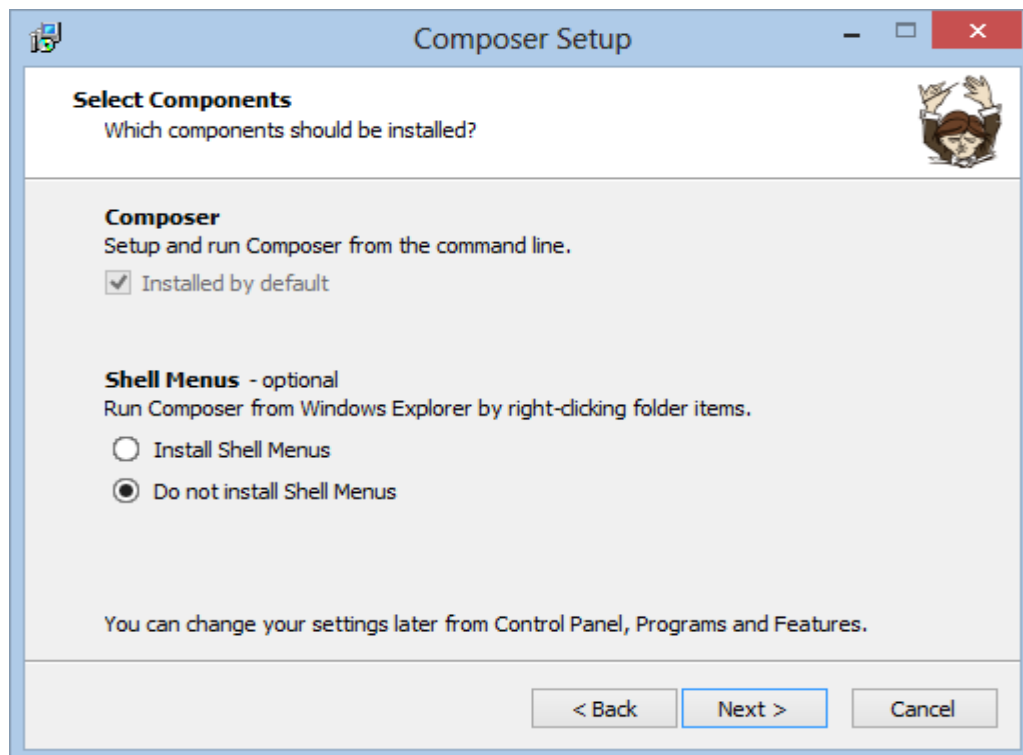


Figura 59 Instalar COMPOSER 2

6. A continuación solicita la ruta del ejecutable de PHP, en este caso como se está trabajando con XAMPP el ejecutable de PHP se encuentra en la ruta: **C:\xampp\php\php.exe**. Posteriormente hacer click en **"Next"**.

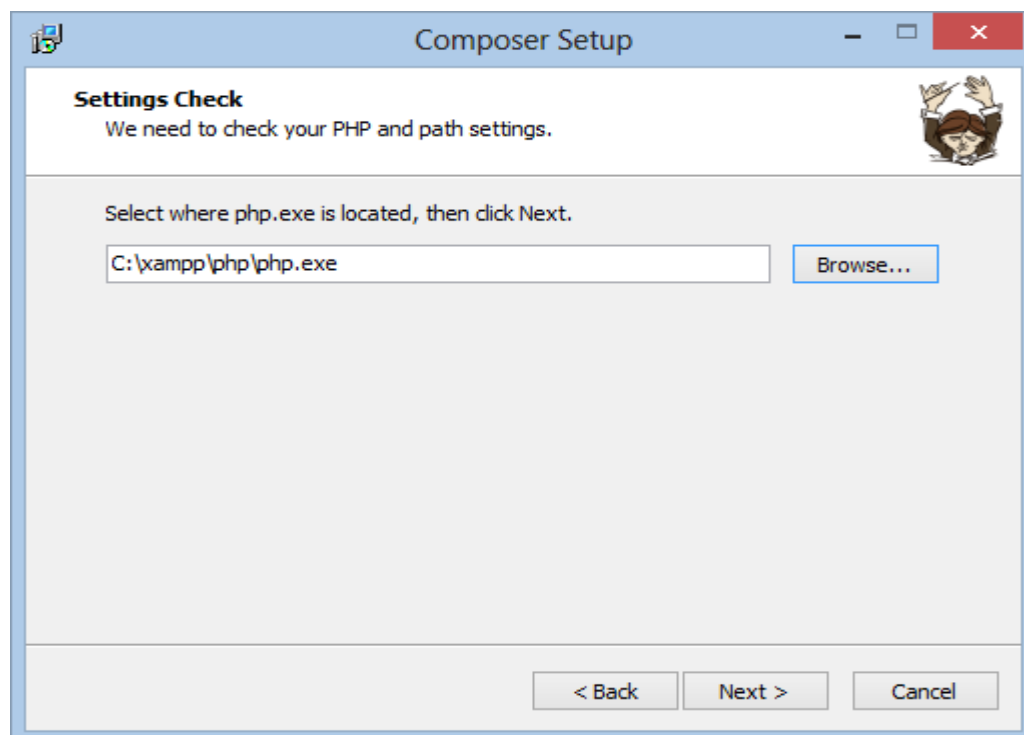


Figura 60 Instalar COMPOSER 3

7. En este punto el instalador muestra la configuración de la instalación, simplemente hay que hacer click a **“Install”**.
8. Después de realizar todo lo anterior se quedará instalado COMPOSER.
9. El siguiente paso es ir a la carpeta del ordenador:

C:\xampp\htdocs.
10. **Se crea una carpeta** con un nombre (por ejemplo tfg5).
11. **Se descomprime** el código de la aplicación dentro de esta carpeta.

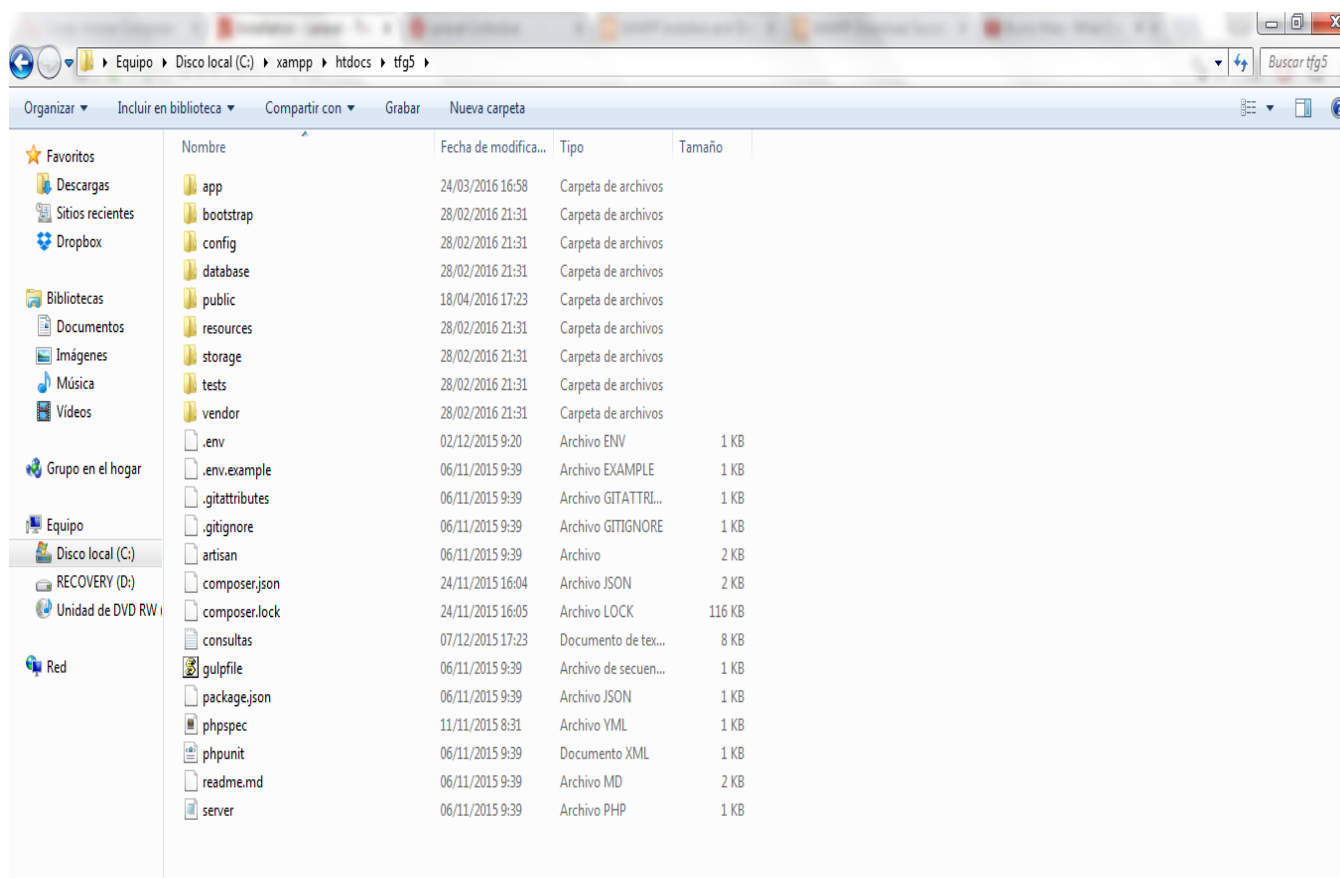


Figura 61 Archivo descomprimido en la carpeta tfg5

12. A continuación hay que ir a la consola de Windows (cmd).
13. Hay que situarse en la ruta **C:\xampp\htdocs\tfg5**.

```
C:\Windows\system32\cmd.exe
02/12/2015 18:48 34.625.032 tfg5_02_12_15.zip
03/12/2015 19:12 34.631.101 tfg5_03_12_15.zip
04/12/2015 11:15 34.657.564 tfg5_04_12_15.zip
05/12/2015 14:18 34.681.774 tfg5_05_12_15.zip
10/12/2015 21:50 34.725.871 tfg5_10_12_15.zip
11/12/2015 15:02 34.733.921 tfg5_11_12_15.zip
12/01/2016 22:47 34.792.879 tfg5_12_1_16.zip
17/12/2015 11:36 34.754.014 tfg5_17_12_15.zip
17/12/2015 17:50 34.773.747 tfg5_17_12_2015.zip
18/12/2015 00:18 34.777.903 tfg5_18_12_15.zip
25/04/2016 15:46 22.343.749 tfg5_25_04.zip
03/12/2015 22:52 34.650.375 tfg5_3_12_15.zip
04/12/2015 13:06 34.662.556 tfg5_4_12_15.zip
05/12/2015 17:42 34.692.189 tfg5_5_12_15.zip
08/12/2015 13:41 34.706.236 tfg5_8_12_15.zip
25/04/2016 15:22 <DIR> tfg6
25/04/2016 15:53 <DIR> tfg7
18/04/2016 09:58 67.388.317 tfg5_000000y.zip
01/12/2015 21:39 <DIR> webalizer
01/12/2015 21:39 <DIR> xampp
34 archivos 1.156.973.759 bytes
11 dirs 253.558.034.432 bytes libres

C:\xampp\htdocs>cd tfg5
C:\xampp\htdocs\tfg5>
```

Figura 62 Consola con la ruta a la carpeta tfg5

14. Se introduce el comando:

- **composer install**

15. Luego el comando:

- **composer update**

16. Se arranca:

- **Apache**
- **MySQL**

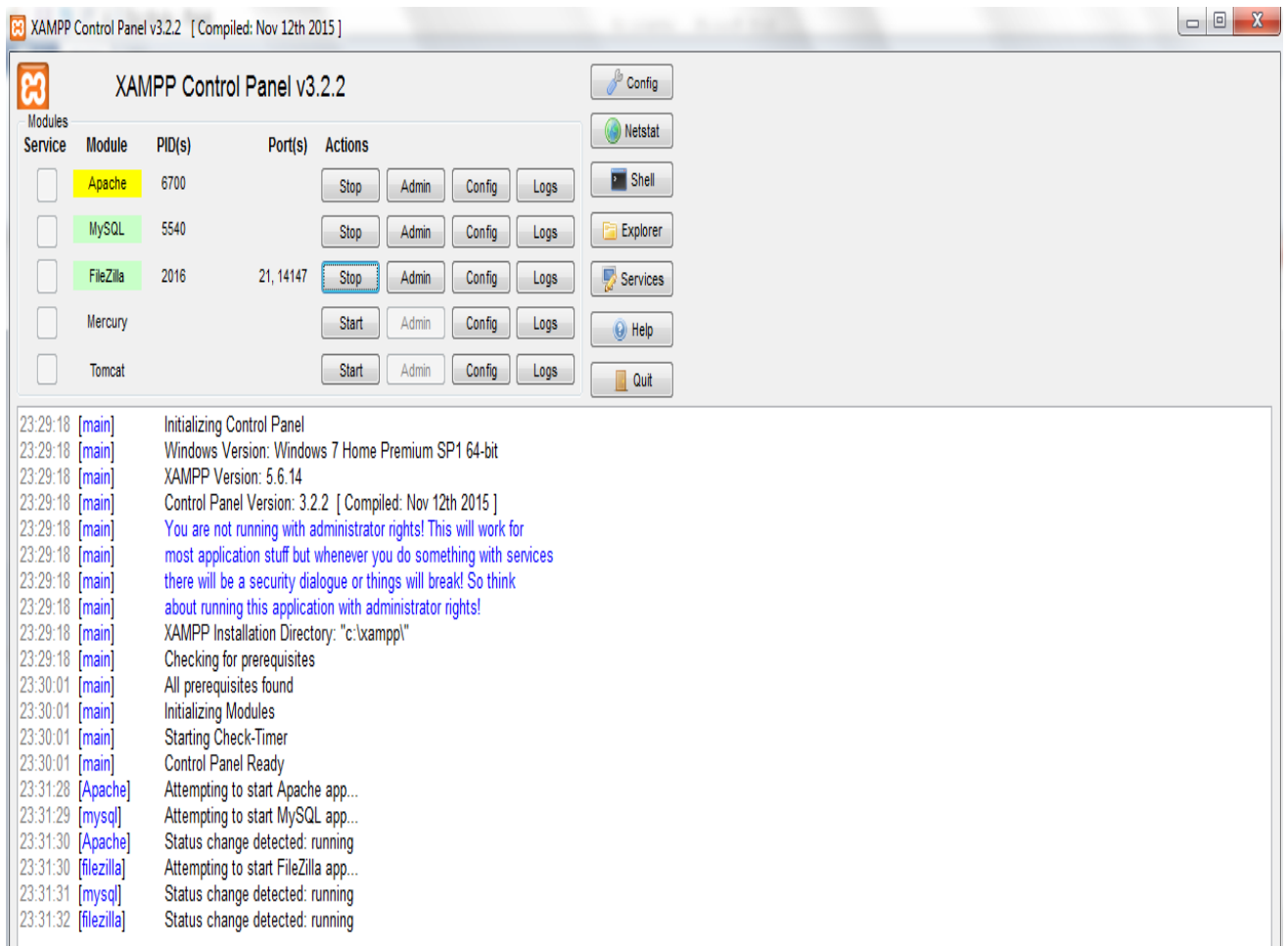


Figura 63 Arrancar Apache y MySQL

17. Hay que abrir un navegador web (preferiblemente Google Chrome) y se introduce en la URL lo siguiente:

- <http://localhost/tfg5/public/login>

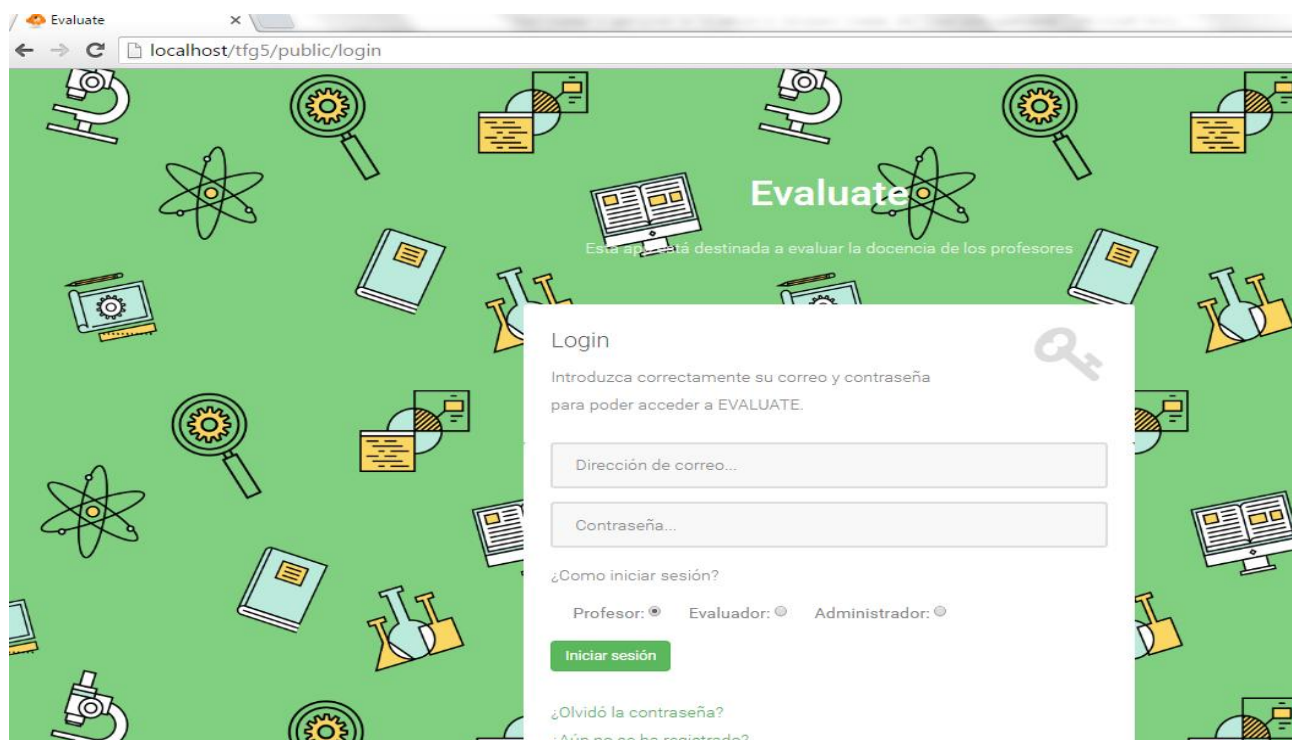


Figura 64 Navegador con `http://localhost/tfg5/public/login` en la URL

18. Con esto termina la instalación.

Anexo 3: Guía de uso de la aplicación

Introducción

La finalidad de este documento es explicar cómo usar la aplicación EVALUATE.

Primero se explica cómo funcionan las partes relacionadas con el usuario y que sirven para los 3 tipos de roles.

Posteriormente se explicará cómo funciona la aplicación teniendo en cuenta los 3 roles que existen.

Guía de uso del Usuario

A continuación se van a explicar las diferentes acciones:

- **Acceder a la aplicación**
- **Registrarse en la aplicación**
- **Recuperar la contraseña**
- **Dar de alta una cuenta**
- **Cerrar sesión**
- **Cambiar foto de perfil**
- **Cambiar contraseña**

Acceder a la aplicación

Para poder acceder a la aplicación hay que abrir un navegador de internet (preferiblemente Google Chrome) e introducir en la URL la siguiente dirección: <http://autoevaluate.esy.es/>

Registrarse en la aplicación

Para poder registrarse en la aplicación hay que pinchar en:

- **¿Aún no se ha registrado?**

Tras realizar este paso aparecerá una nueva vista hay que rellenar todo los campos.

Una vez rellenado todos estos campos hay que hacer click en el botón:

- **Crear cuenta**

Si se han completado todos estos pasos correctamente, se recibirá una notificación de que ha sido registrado.

Recuperar la contraseña

Para poder recuperar la contraseña hay que pinchar en:

- **¿Olvidó la contraseña?**

Tras realizar este paso aparecerá una nueva vista, hay que introducir el correo electrónico de la cuenta que desea recuperar la contraseña.

Una vez indicado el correo electrónico hay que hacer click en el botón:

- **Recuperar contraseña**

Posteriormente hay que ir al correo electrónico y abrir el mensaje que ha enviado Evaluate.

A continuación hay que pinchar en el enlace que aparece en el correo. Al hacer click en este enlace aparecerá una vista como la figura siguiente:



The image shows a web form for password recovery. The title is "¿Olvidaste la contraseña?". Below the title, it says "Introduce la información que se solicita a continuación para poder recuperar tu contraseña." There are three radio buttons for role selection: "Profesor:" (selected), "Evaluador:", and "Administrador:". Below these are three input fields: "Dirección de correo...", "Contraseña...", and "Repita la contraseña...". At the bottom is a green button labeled "Restablecer la contraseña". The background is green with various educational icons like a microscope, atom, and books.

Figura 65 Recuperar contraseña - ¿Olvido la contraseña?

Hay que rellenar el formulario que aparece.

Tras completar el formulario hay que pinchar en el botón:

- **Restablecer la contraseña**

De esta manera se habrá recuperado la contraseña.

Dar de alta una cuenta

Para dar de alta una cuenta que ha sido creada previamente por el administrador, hay que ir al correo electrónico y abrir el mensaje que ha enviado Evaluate.

En este correo hay que pinchar en el enlace que aparece.

Tras pinchar en el enlace aparecerá una nueva pantalla como la de la figura.

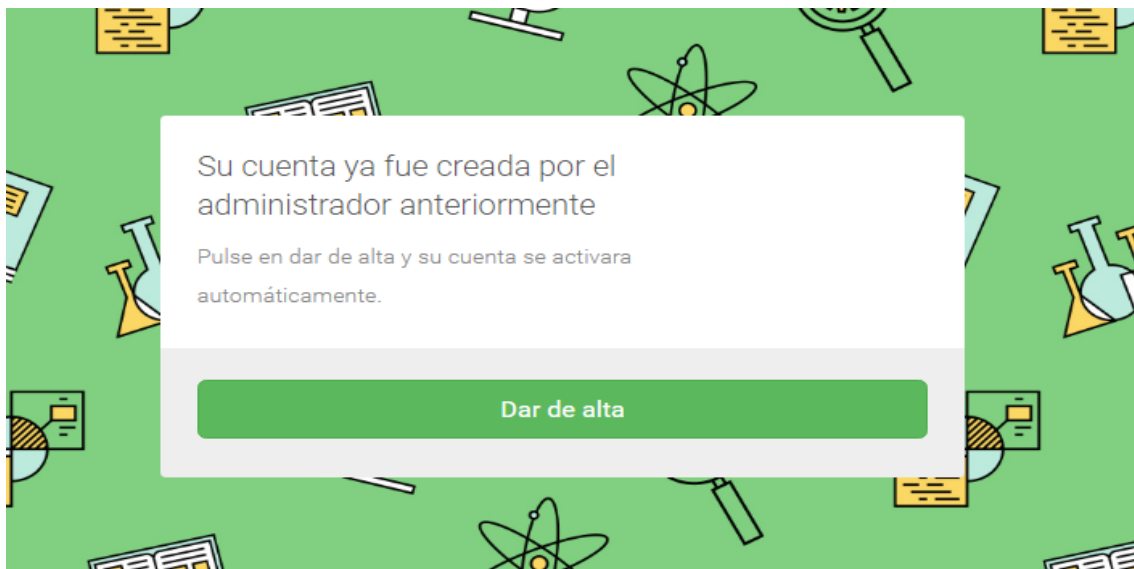


Figura 66 Dar de alta una cuenta

A continuación hay que pinchar en el botón:

- **Dar de alta**

Una vez realizado este paso, la cuenta habrá sido activada.

Cerrar sesión

Para cerrar sesión hay que ir a la foto de perfil y al nombre (situado arriba del todo a la derecha) y pinchar o bien en la imagen o bien en el nombre.

Una vez realizado el paso anterior se desplegará un menú hay que pinchar en:

- **Cerrar sesión**

Una vez realizado este proceso, se cerrará la sesión.

Cambiar foto de perfil

Para cambiar de foto de perfil hay que ir a la foto de perfil y al nombre (situado arriba del todo a la derecha) y pinchar o bien en la imagen o bien en el nombre.

Se desplegará un menú hay que pinchar en:

- **Ver Perfil**

A continuación aparece otra pantalla que muestra la foto de perfil más grande.

Justo debajo en el botón:

- **Seleccionar archivo**, hay que seleccionar la nueva foto.

Posteriormente se tiene que pinchar en el botón:

- **Cambiar de foto de perfil**

De esta forma se cambiará la foto de perfil.

Cambiar contraseña

Este proceso es el mismo para los tres roles: administrador profesor y evaluador.

Para cambiar la contraseña hay que ir a la foto de perfil y al nombre (situado arriba del todo a la derecha) y pinchar o bien en la imagen o bien en el nombre.

Se desplegará un menú hay que pinchar en:

- **Ver Perfil**

A continuación aparecerá una nueva pantalla, en la parte inferior de dicha pantalla hay que pinchar en:

- **Cambiar contraseña**

Después de pinchar aparece un formulario que hay que rellenar.

Una vez completado, hay que pinchar en el botón:

- **Guardar cambios**

Si todo hay ido correctamente, la contraseña se habrá cambiado.

Guía de uso del rol profesor

A continuación se van a explicar las diferentes acciones:

- **Borrar una asignatura**
- **Borrar una asignatura definitivamente.**
- **Recuperar una asignatura borrada anteriormente.**
- **Evaluar una asignatura manualmente.**
- **Evaluar una asignatura automáticamente.**
- **Ver una evaluación.**
- **Descargar el resultado en PDF.**
- **Comparar el resultado de una asignatura con otra.**
- **Modificar el resultado de la evaluación de una asignatura.**
- **Continuar con la evaluación de una asignatura.**

Borrar una asignatura

Para borrar una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas disponibles**

Una vez realizado este paso, aparece una nueva vista con un listado de asignaturas, si es que las hubiese.

Para borrar la asignatura hay que pinchar en el botón:

- **Eliminar asignatura**

De esta forma la asignatura quedará eliminada.

Borrar una asignatura definitivamente

Para borrar una asignatura definitivamente hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas eliminadas**

Una vez realizado este paso, aparece una nueva vista con un listado de asignaturas, si es que las hubiese.

Para borrar definitivamente la asignatura hay que pinchar en el botón:

- **Eliminar definitivamente**

De esta manera la asignatura se habrá eliminado definitivamente.

Recuperar una asignatura borrada anteriormente

Para recuperar una asignatura borrada hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas eliminadas**

Una vez realizado este paso, aparece una nueva vista con un listado de asignaturas, si es que las hubiese.

Para recuperar la asignatura hay que pinchar en el botón:

- **Deshacer eliminación**

De esta forma se recupera una asignatura.

Evaluar una asignatura manualmente

Para evaluar una asignatura manualmente hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias**

A continuación de despliega un menú hay que pinchar en:

- **Asignaturas no evaluadas**

Tras realizar estos pasos aparece una lista de asignaturas.

Hay que pinchar en el botón:

- **Evaluar**

Posteriormente aparece otra página hay que pinchar en el botón:

- **Evaluar manualmente**

A continuación hay que rellenar el cuestionario y pulsar el botón que corresponda:

- **Guardar cambios sin terminar evaluación** o
- **Guardar cambios y terminar evaluación**

Evaluar una asignatura automáticamente

Para evaluar una asignatura manualmente hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias**

A continuación se despliega un menú hay que pinchar en:

- **Asignaturas no evaluadas**

Tras realizar estos pasos aparece una lista de asignaturas.

Hay que pinchar en el botón:

- **Evaluar**

Aparece otra página y hay que **subir dos** ficheros:

- **Boletín de notas**
- **Logs del Moodle**

Hay que **seleccionar la versión de** Moodle que se está utilizando.

Una vez realizado los pasos anteriores hay que pinchar en el botón:

- **Evaluar automáticamente**

A continuación hay que rellenar el cuestionario y pulsar el botón correspondiente:

- **Guardar cambios sin terminar evaluación** o
- **Guardar cambios y terminar evaluación**

Ver una evaluación

Para poder ver la evaluación de una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias** o
- **Evaluaciones ajenas**

La elección escogida dependerá de si se quiere ver una evaluación hecha por el propio usuario o por un evaluador.

Al pinchar sobre estos botones se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Tras realizar estos pasos aparece una lista de asignaturas (si existen asignaturas evaluadas), hay que pinchar en el botón:

- **Ver evaluación**

De esta forma se ve una evaluación.

Descargar el resultado en PDF

Para poder descargar la evaluación de una asignatura en PDF hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias** o
- **Evaluaciones ajenas**

La elección escogida dependerá de si se quiere ver una evaluación hecha por el propio usuario o por un evaluador.

Al pinchar sobre estos botones se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Posteriormente aparece una lista de asignaturas (si existen asignaturas evaluadas), hay que pinchar en el botón:

- **Descargar pdf**

Así se descarga el resultado en PDF.

Comparar el resultado de una asignatura con otra

Para poder comparar una asignatura con otra hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias** o
- **Evaluaciones ajenas**

La elección escogida dependerá de si se quiere ver una evaluación hecha por el propio usuario o por un evaluador.

Al pinchar sobre estos botones se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Tras realizar estos pasos aparece una lista con las asignaturas evaluadas (si existen asignaturas), hay que pinchar en el botón:

- **Comparar**

Después aparece una lista de asignaturas (si existen asignaturas para comparar) hay que seleccionar con que asignatura se va a comparar y pinchar en el botón:

- **Seleccionar**

De esta forma se puede comparar una asignatura con otra.

Modificar el resultado de la evaluación de una asignatura

Para poder modificar una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Tras realizar estos pasos aparece una lista de asignaturas (si existen asignaturas evaluadas), hay que pinchar en el botón:

- **Modificar**

De esta forma se modificará una asignatura.

Continuar con la evaluación de una asignatura

Para poder continuar con la evaluación de una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones propias**

Al pinchar sobre estos botones se despliega un submenú, hay que pinchar en:

- **Asignaturas en proceso de evaluación**

Tras realizar estos pasos aparece una lista de asignaturas (si existen asignaturas para seguir evaluando), hay que pinchar en el botón:

- **Continuar evaluando**

De esta forma se continúa con la evaluación de una asignatura.

Guía de uso del rol evaluador

A continuación se van a explicar las diferentes acciones:

- **Solicitar asignatura para evaluar.**
- **Borrar una asignatura.**
- **Borrar una asignatura definitivamente.**
- **Recuperar una asignatura borrada anteriormente.**
- **Evaluar una asignatura manualmente.**
- **Evaluar una asignatura automáticamente.**
- **Ver una evaluación.**
- **Descargar el resultado en PDF.**
- **Comparar el resultado de la evaluación una asignatura con otra.**
- **Modificar el resultado de la evaluación de una asignatura.**
- **Continuar con la evaluación de una asignatura.**

Solicitar asignatura para evaluar

Para poder solicitar una asignatura para evaluar hay que ir al menú lateral izquierdo y pinchar en:

- **Solicitar asignaturas para evaluar**

Tras realizar este paso aparece una lista de asignaturas que puede solicitar (si existen asignaturas para poder solicitar), hay que pinchar en el botón:

- **Solicitar asignatura**

De esta forma se habrá solicitado una asignatura.

Borrar una asignatura

Para borrar una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Lista de asignaturas**

Una vez realizado este paso, aparece una nueva vista con un listado de asignaturas (en el caso de tener asignaturas).

Para borrar la asignatura hay que pinchar en el botón:

- **Eliminar asignatura**

De esta forma se habrá eliminado una asignatura

Borrar una asignatura definitivamente

Para borrar una asignatura definitivamente hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas eliminadas**

Una vez realizado este paso, aparece una nueva vista con un listado de asignaturas (en el caso de tener asignaturas que han sido eliminadas).

Para borrar definitivamente la asignatura hay que pinchar en el botón:

- **Eliminar definitivamente**

De esta forma se habrá eliminado definitivamente una asignatura

Recuperar una asignatura borrada anteriormente

Para recuperar una asignatura borrada hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas eliminadas**

Una vez realizado este paso, aparece una nueva vista con un listado de asignaturas (en el caso de tener asignaturas que han sido eliminadas).

Para recuperar la asignatura hay que pinchar en el botón:

- **Deshacer eliminación**

De esta forma se habrá recuperado una asignatura.

Evaluar una asignatura manualmente

Para evaluar una asignatura manualmente hay que ir al menú lateral izquierdo y pinchar en:

- **Todas las asignaturas**

A continuación se despliega un menú, hay que pinchar en:

- **Asignaturas no evaluadas**

Tras realizar estos pasos aparece una lista de asignaturas (en el caso de que tengamos asignaturas).

Hay que pinchar en el botón:

- **Evaluar**

Después de pinchar en este botón aparece otra página, hay que pinchar en el botón:

- **Evaluar manualmente**

A continuación hay que rellenar el cuestionario y pulsar en el botón:

- **Guardar cambios sin terminar evaluación** o
- **Guardar cambios y terminar evaluación**, dependiendo de lo que desee.

Evaluar una asignatura automáticamente

Para evaluar una asignatura manualmente hay que ir al menú lateral izquierdo y pinchar en:

- **Todas las asignaturas**

A continuación se despliega un menú, hay que pinchar en:

- **Asignaturas no evaluadas**

Tras realizar estos pasos nos aparece una lista de asignaturas (en el caso de que tengamos asignaturas).

Hay que pinchar en el botón:

- **Evaluar**

Después de pinchar en este botón aparece otra página, hay que **subir dos** ficheros:

- **Boletín de notas**
- **Logs del Moodle**

Hay que **seleccionar la versión de Moodle** que se está utilizando.

Una vez realizado los pasos anteriores hay que pinchar en el botón:

- **Evaluar automáticamente**

A continuación hay que rellenar el cuestionario y pulsar el botón correspondiente dependiendo de lo que se desee:

- **Guardar cambios sin terminar evaluación** o
- **Guardar cambios y terminar evaluación**

Ver una evaluación

Para poder ver la evaluación de una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Todas las asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Después de realizar estos pasos aparece una lista de asignaturas (si existen asignaturas evaluadas), hay que pinchar en el botón:

- **Ver evaluación**

De esta forma se ve una evaluación.

Descargar el resultado en PDF

Para poder descargar la evaluación de una asignatura en PDF hay que ir al menú lateral izquierdo y pinchar en:

- **Todas las asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Después de realizar estos pasos aparece una lista de asignaturas (si existen asignaturas evaluadas), hay que pinchar en el botón:

- **Descargar pdf**

De esta manera se puede descargar el resultado en PDF.

Comparar el resultado de la evaluación de una asignatura con otra

Para poder comparar una asignatura con otra hay que ir al menú lateral izquierdo y pinchar en:

- **Todas las asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

A continuación aparece una lista con las asignaturas que podemos comparar (si existen asignaturas para comparar), hay que pinchar en el botón:

- **Comparar**

De esta forma se compara una asignatura con otra.

Modificar el resultado de la evaluación de una asignatura

Para poder modificar una asignatura hay que ir al menú lateral izquierdo y pinchar en pinchar en:

- **Todas las asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Asignaturas evaluadas**

Después de realizar estos pasos aparece una lista de asignaturas (si existen asignaturas evaluadas), hay que pinchar en el botón:

- **Modificar**

De esta forma se modifica una asignatura.

Continuar con la evaluación de una asignatura

Para poder continuar con la evaluación de una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Todas las asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Asignaturas en proceso de evaluación**

Después de realizar estos pasos aparece una lista de asignaturas (si existen asignaturas para seguir evaluando), hay que pinchar en el botón:

- **Continuar evaluando”**

De esta forma se continúa con la evaluación de una asignatura.

Guía de uso del rol administrador

A continuación se van a explicar las diferentes acciones:

- **Aceptar solicitud de un usuario.**
- **Rechazar solicitud de un usuario.**
- **Aceptar solicitud de una asignatura para ser evaluada por un evaluador.**
- **Rechazar solicitud de una asignatura para ser evaluada por un evaluador.**
- **Eliminar un usuario.**
- **Modificar un usuario.**
- **Agregar un usuario manualmente.**
- **Agregar usuarios mediante la subida de un fichero.**
- **Eliminar una asignatura.**
- **Agregar una asignatura manualmente.**
- **Agregar asignaturas mediante la subida de un fichero.**
- **Agregar un formulario.**
- **Agregar criterios de evaluación.**
- **Eliminar una versión Moodle.**
- **Agregar una versión Moodle.**
- **Ver evaluación o descargar resultado en PDF**
- **Asignar evaluador a una asignatura**

Aceptar solicitud de un usuario

Para poder aceptar la solicitud de un usuario hay que ir al menú lateral izquierdo y pinchar en:

- **Solicitudes de cuentas nuevas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Solicitudes de profesores** o
- **Solicitudes de evaluadores** o
- **Solicitudes de administradores**

Dependiendo del rol elegido.

Después de realizar estos pasos aparece una lista de asignaturas (si existen solicitudes), hay que pinchar en el botón:

- **Aceptar solicitud**

De esta forma se acepta la solicitud de un usuario.

Rechazar solicitud de un usuario

Para poder rechazar la solicitud de un usuario hay que ir al menú lateral izquierdo y pinchar en:

- **Solicitudes de cuentas nuevas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Solicitudes de profesores** o
- **Solicitudes de evaluadores** o
- **Solicitudes de administradores**

Dependiendo del rol elegido.

Después de realizar estos pasos aparece una lista de asignaturas (si existen solicitudes), hay que pinchar en el botón:

- **Rechazar solicitud**

De esta forma se rechaza una solicitud de un usuario.

Aceptar solicitud de una asignatura para ser evaluada por un evaluador

Para poder aceptar la solicitud de una asignatura para ser evaluada por un evaluador, hay que ir al menú lateral izquierdo y pinchar en:

- **Solicitudes de asignaturas**

Después de realizar este paso aparece una lista de asignaturas (si existen solicitudes), hay que pinchar en el botón:

- **Aceptar solicitud**

De esta forma se acepta la solicitud de una asignatura para ser evaluada por un evaluador.

Rechazar solicitud de una asignatura para ser evaluada por un evaluador

Para poder rechazar la solicitud de una asignatura para ser evaluada por un evaluador, hay que ir al menú lateral izquierdo y pinchar en:

- **Solicitudes de asignaturas**

Después de realizar este paso aparece una lista de asignaturas (si existen solicitudes), hay que pinchar en el botón:

- **Rechazar solicitud**

De esta forma se rechaza la solicitud de una asignatura para ser evaluada por un evaluador.

Eliminar un usuario

Para poder eliminar un usuario hay que ir al menú lateral izquierdo y pinchar en:

- **Profesores** o
- **Evaluadores** o
- **Administradores**

Dependiendo del rol del usuario que se desea borrar.

Al pinchar sobre estos botones se despliega un submenú, hay que pinchar en:

- **Lista de profesores** o
- **Lista de evaluadores** o
- **Lista de administradores**

Después de realizar estos pasos aparece una lista de usuarios (si existen usuarios con ese rol), hay que pinchar en el botón:

- **Eliminar**

De esta forma se elimina a un usuario.

Modificar un usuario

Para poder modificar un usuario se debe ir al menú lateral izquierdo y pinchar en:

- **Profesores** o
- **Evaluadores** o
- **Administradores**

Dependiendo del rol del usuario que se desea borrar.

Al pinchar sobre estos botones se desplegara un submenú donde se puede pinchar en:

- **Lista de profesores** o
- **Lista de evaluadores** o
- **Lista de administradores**

Tras realizar estos pasos aparecerá una lista de usuarios (si existen usuarios con ese rol), se debe pinchar en el botón:

- **Modificar**

De esta forma se modificará usuario.

Agregar un usuario manualmente

Para poder agregar un usuario manualmente hay que ir al menú lateral izquierdo y pinchar en:

- **Profesores"** o
- **Evaluadores** o
- **Administradores**

Dependiendo del rol del usuario que se desea agregar.

Al pinchar sobre estos botones se despliega un submenú, se puede pinchar en:

- **Agregar profesores** o
- **Agregar evaluadores** o
- **Agregar administradores**

Después de realizar estos pasos aparece una nueva vista, hay que pinchar en el botón:

- **Manual**

Por último, hay que rellenar un formulario y pinchar en el botón:

- **Crear profesor** o
- **Crear evaluador** o
- **Crear administrador**

Dependiendo de la vista en la que nos encontremos.

De esta forma se crea un usuario manualmente.

Agregar usuarios mediante la subida de un fichero

Para poder agregar un usuario mediante la subida de un fichero hay que ir al menú lateral izquierdo y pinchar en:

- **Profesores** o
- **Evaluadores** o
- **Administradores**

Dependiendo del rol del usuario que se desea agregar.

Al pinchar sobre estos botones se despliega un submenú, hay que pinchar en:

- **Agregar profesores** o
- **Agregar evaluadores** o
- **Agregar administradores**

Después de realizar estos pasos aparece una nueva vista, hay que pinchar en el botón:

- **Subida de un fichero**

Por último, hay que subir un fichero con la estructura explicada y tras realizar este paso hay que pinchar en el botón:

- **Subir fichero profesor** o
- **Subir fichero evaluador** o
- **Subir fichero administrador**

Dependiendo de la vista donde nos encontremos.

De esta forma se crea usuarios mediante la subida de un fichero.

Eliminar una asignatura

Para poder eliminar una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en

- **Lista asignaturas**

Después de realizar estos pasos aparece una lista de asignaturas (si existen asignaturas), hay que pinchar en el botón:

- **Eliminar**

De esta forma se elimina una asignatura.

Agregar una asignatura manualmente

Para poder agregar una asignatura manualmente hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Agregar asignatura**

Después de realizar estos pasos aparece una nueva vista, hay que pinchar en el botón:

- **Manual**

Por último, hay que rellenar un formulario y pinchar en el botón:

- **Agregar asignatura**

De esta forma se crea una asignatura manualmente.

Agregar asignaturas mediante la subida de un fichero

Para poder agregar asignaturas mediante la subida de un fichero hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Agregar asignatura**

Después de realizar estos pasos aparece una nueva vista, hay que pinchar en el botón:

- **Subida de un fichero**

Por último, hay que subir un fichero con la estructura explicada y tras realizar este paso hay que pinchar en el botón

- **Subir**

De esta forma se crean asignaturas mediante la subida de un fichero.

Agregar un formulario

Para poder agregar un formulario hay que ir al menú lateral izquierdo y pinchar en:

- **Configuración**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Formulario**

Después de realizar estos pasos aparece una nueva vista, hay que subir un fichero con la estructura adecuada.

Una vez que se ha subido el fichero hay que pinchar en el botón:

- **Subir**

De esta forma se agrega un nuevo formulario.

Agregar criterios de evaluación

Para poder agregar criterios de evaluación hay que ir al menú lateral izquierdo y pinchar en:

- **Configuración**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Criterios de evaluación**

Después de realizar estos pasos aparece una nueva vista, hay que subir un fichero con la estructura explicada anteriormente.

Una vez que se ha subido el fichero hay que pinchar en el botón:

- **Subir**

De esta forma se agregan los criterios de evaluación.

Eliminar una versión Moodle

Para poder eliminar una versión Moodle hay que ir al menú lateral izquierdo y pinchar en:

- **Configuración**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Versiones Moodle**

Después de realizar este último paso volverá a aparecer un submenú, hay que pinchar en:

- **Lista de versiones**

Tras realizar estos pasos aparecerá una lista de versiones Moodle (si existen), hay que pinchar en el botón:

- **Eliminar**

De esta forma se elimina una versión Moodle.

Agregar una versión Moodle

Para poder agregar una versión Moodle hay que ir al menú lateral izquierdo y pinchar en:

- **Configuración**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en:

- **Versiones Moodle**

Después de realizar este paso volverá a aparecer un submenú, hay que pinchar en:

- **Agregar versión**

Tras realizar estos pasos aparecerá un formulario que hay que rellenar y a continuación pinchar en el botón:

- **Agregar versión Moodle**

De esta forma se agrega una nueva versión Moodle.

Ver evaluación o descargar resultado en PDF

Para poder ver una evaluación o descargar el resultado en PDF hay que ir al menú lateral izquierdo y pinchar en:

- **Evaluaciones**

Al pinchar sobre este botón se despliega un submenú hay que elegir que evaluación se quiere ver: de profesor o de evaluador.

A continuación aparece una lista con las evaluaciones (si hay evaluaciones), hay que pinchar en el botón:

- **Ver evaluación**
- **Descargar PDF**

Dependiendo de la acción que desee realizar.

Asignar evaluador a una asignatura

Para poder asignar un evaluador a una asignatura hay que ir al menú lateral izquierdo y pinchar en:

- **Asignaturas**

Al pinchar sobre este botón se despliega un submenú, hay que pinchar en el botón:

- **Lista Asignaturas**

A continuación aparece una lista con las asignaturas (si hay asignaturas), hay que pinchar en el botón:

- **Asignar evaluador**

Después aparece una lista de evaluadores (si hay evaluadores), hay que seleccionar el evaluador que se desee asignar a la asignatura pinchando en el botón:

- **Asignar**

De esta forma se asigna un evaluador a una nueva asignatura.

Glosario

HTML: Hypertext Markup Language

Lenguaje de marcado para el desarrollo de aplicaciones web, definiendo una estructura base para la distribución de la información a mostrar a los usuarios.

HTTP: Hypertext Transfer Protocol

Protocolo empleado para transacciones y comunicación con la parte del servidor de una aplicación web mediante peticiones, para tratar la información solicitar recursos, etc.

CSS: Cascading Style Sheets

Lenguaje para definir el diseño de un archivo estructurado mediante un lenguaje de marcado como es *HTML* o *XML*.

JSON: JavaScript Object Notation

Formato ligero para el intercambio de datos, su empleo es debido a su simplicidad.

PHP: Hypertext Preprocessor

Lenguaje de programación de uso de código al lado del servidor para el desarrollo de aplicaciones web dinámicas.

AJAX: Asynchronous JavaScript And XML

Es una técnica de desarrollo web para crear aplicaciones interactivas.

SQL: Structured Query Language

Lenguaje declarativo de acceso a bases de datos relacionales a través de lógica relación.

BD: Base de datos

Banco de información con datos relacionados por diversas temáticas siendo resuelta mediante loca relacional.

WAMPP: Windows, Apache, MySQL, PHP

Permite gestionar páginas web además de datos sobre ellas, proporcionando lenguajes para su desarrollo para Windows mediante Apache, MySQL y PHP.

URL: Uniform Resource Locator

Identificador de recursos uniforme, cuyos recursos de la red a los que apunta pueden variar.

MVC: Modelo Vista Controlador

Es un modelo de arquitectura software cuya finalidad es la separación de datos interfaz y lógica de control aumentando la claridad del código y que es lo que se está tratando en cada momento.

LOG Moodle

Registro oficial de eventos durante un rango de tiempo en particular. Se encarga de registrar datos e información correspondiente al Moodle.

Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP.

Bootstrap

Permite el un desarrollo front-end web más rápido y sencillo, ajustando el diseño de las aplicaciones webs a diferentes resoluciones.

Moodle

Aplicación web educativa que contiene un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Framework

Desde el punto de vista de desarrollo software, es una estructura de soporte en la cual se pueden desarrollar otros proyectos software, por lo que actúa como una aplicación genérica.

Servidor web

Parte de una aplicación web en la que se realiza las conexiones con el cliente y a la que se le piden recursos, tratamiento de datos y servicios solicitados con las interacciones de los usuarios.

Cliente web

Parte de una aplicación web, la cual captura las interacciones de los usuarios y son tratadas en el lado del servidor mediante tratamiento de la información.

MongoDB

Sistema de base de datos no relacional orientado a documentos tipo JSON bajo código abierto. Las consultas sobre ellas son más rápidas debido a la ausencia de lógica relación, aunque pierde consistencia e integridad de los datos almacenados.

Tablets

Dispositivo portátil táctil de entre 6 y 12 pulgadas.

Portátil

Dispositivo fácil de transportar pesando entre 1 y 3 kg, mayormente para referirse a ordenadores portátiles.

Smartphones

Dispositivo móvil que dispone de hardware y sistema operativo permitiendo la realización de tareas similares a las de un ordenador, incluso las funcionalidades propias de un teléfono móvil.

Click

Consiste en pulsar el botón izquierdo del ratón del ordenador.

Script

Archivo con órdenes o rutas.

CodeIgniter

Framework PHP para la creación rápido de aplicaciones web de código abierto.

Spring

Framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

Symfony

Framework de código abierto centrado en el desarrollo de aplicaciones en lado de

servidor, permitiendo su desarrollo en menor tiempo.

DOMPDF

Convertidor de código HTML a un archivo en PDF.

Eloquent ORM

Sistema de mapeo de datos relacional, que facilita la creación de modelos abstrayendo de la implementación interna.

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Window

Sistema operativo de Microsoft.

Linux

Sistema operativo de código abierto.